mathematics

MDPI

Article

A DNA Encoding Image Encryption Algorithm Based on Chaos

Li Huang [1,2], Cong Ding [1,2,*], Zhenjie Bao [1,2], Haitao Chen [3] and Changsheng Wan [1,2]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China
[2] Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200444, China
[3] Hunan Zhentong Zhiyong Artificial Intelligence Technology Co., Ltd., Changsha 410026, China
* Correspondence: dingcong@seu.edu.cn

Abstract: With the development of society and the Internet, images have become an important medium for information exchange. To improve the security of image encryption and transmission, a new image encryption algorithm based on bit-plane decomposition, DNA encoding and the 5D Hamiltonian conservative chaotic system is proposed. This encryption scheme is different from the traditional scrambling and diffusion methods at the level of image spatial pixels but encodes images into DNA strands and completely scrambles and diffuses operations on the DNA strands to ensure the security of images and improve the efficiency of image encryption. Firstly, the initial value sequence and convolution kernel of the five-dimensional Hamiltonian conservative chaotic system are obtained using SHA-256. Secondly, the bit-plane decomposition is used to decompose the image into high-bit and low-bit-planes, combine with DNA encoding to generate DNA strands, hide the large amount of valid information contained in the high-bit-planes, and preliminarily complete the hiding of the image information. In order to further ensure the effect of image encryption, seven DNA operation index tables controlling the diffusion process of the DNA strands are constructed based on the DNA operation rules. Finally, the scrambled and diffused DNA strand is decomposed into multiple bit-planes to reconstruct an encrypted image. The experimental results and security analysis show that this algorithm has a large enough key space, strong key sensitivity, high image encryption quality, strong robustness and high encryption efficiency. In addition, it can resist statistical attacks, differential attacks, and common attacks such as cropping attack, noise attack and classical attack.

Keywords: image encryption; DNA encoding; conservative chaotic system; DNA strand

MSC: 68U10

check for updates

Academic Editor: Janez Žerovnik

Received: 12 March 2025
Revised: 10 April 2025
Accepted: 15 April 2025
Published: 18 April 2025

Citation: Huang, L.; Ding, C.; Bao, Z.; Chen, H.; Wan, C. A DNA Encoding Image Encryption Algorithm Based on Chaos. *Mathematics* **2025**, *13*, 1330. https://doi.org/10.3390/math13081330

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the popularization of the Internet and various information technologies, digital images carry more effective information and have stronger readability than text. At the same time, digital images play an important role as information carriers in military, medical, and other sensitive fields of national security. Therefore, if the image is stolen or even tampered with by criminals in the transmission process, it is easy to cause the leakage of all kinds of information, which poses a threat to the life and property of citizens and even national security. To prevent images from being stolen or tampered with without authorization, image encryption is an effective method to protect image content security.

The chaotic system has a wide range of applications in image encryption and other cryptographic systems due to the complex dynamic behavior, which greatly improves

the key security in the cryptographic system. As early as 1963, Lorenz discovered the first chaotic attractor through the study of meteorology, which was later known as the Lorenz system [1]. In 1989, Tufillaro further extended the study of the chaotic behavior of chaotic systems by studying the nonlinear vibration phenomenon of elastic strings [2]. The chaotic system is periodic and deterministic, so it has a strong fit with the encryption strategy of a cryptosystem. In the same year, Matthews first proposed the application of chaotic systems to image encryption and using a chaotic system to generate a set of random numbers to ensure the security of keys in the cryptosystem [3]. In 1998, Fridrich first proposed to apply the 2D chaotic system to image encryption, which greatly improves the security of image encryption. Since then, image encryption has gone through a series of developments and derived a series of new image encryption algorithms, such as DNA encoding [4], genetic recombination [5], DWT transforms [6], wavelet transforms [7], compression sensing [8–10], etc. These algorithms are essentially upgrades and improvements of cryptosystems combined with a chaotic system. It has become the main trend of image encryption algorithm research to develop a new image encryption algorithm based on a chaotic system and other technologies. In recent years, many novel and efficient image encryption algorithms have been proposed. The algorithm [11] extended the quantum coding to the image scrambling and diffusion operations, and it used the exchange operation to change the gray value of the image based on the quantum image model. In the same year, the author of [12] combined the Sine map and the Tent map to design a new two-dimensional chaotic system, so as to enhance its chaotic behavior. In 2021, Hua et al. optimized the whole process of sparsifiction, the sampling and embedding of compressed sensing to solve the problems of poor quality of reconstructed images and low efficiency of image encryption. The adaptive thresholding sparsification, the random-order Bernoulli random matrix and the matrix coding technology were introduced to improve the image encryption effect and decryption quality [13]. The S-box is also a relatively new research direction. Hua et al. used the enhanced logistic map as the main core to generate the Latin square matrix to construct the S-box [14]. In the next year, Zou et al. designed a scrambling and diffusion method based on DNA strand, which achieves the purpose of DNA strand scrambling by replacing different DNA strand fragments [15]. Recently, the authors of [16] proposed a new chaotic image compression encryption scheme. In the paper, they combine the filters and feed-forward structure to create a new uniform chaotic system, which has great performance on the generation of key stream. In general, with the in-depth study of chaotic phenomena, the introduction of various other new technologies can be well integrated with chaotic systems. However, the traditional chaotic systems are not invulnerable; they also have a series of security risks, such as being vulnerable to refactoring attacks, as well as having a periodic window and poor sequential randomness [17]. Chaotic systems can be divided into dissipative chaotic systems and conservative chaotic systems according to whether energy is conserved or not. Compared with the dissipative system, the conservative system does not produce attractors, so it has better ergodic property and can resist reconstruction attacks [18]. Compared with the dissipative system, the conservative chaotic system is a more ideal random sequence generator. The authors of [19] draw on this 5D conservative chaotic system combined with the neural network, showing better image encryption quality.

DNA encoding is used to convert digital images into DNA strands by simulating the composition rules of bases in DNA strands. Due to the compatibility between DNA strand and cryptography, a large number of researchers have invested in the research of the image encryption algorithm based on DNA coding. Sakshi et al. designed a symmetric key image encryption algorithm based on the 3D chaotic system and DNA coding; the chaotic system is used to generate the symmetric image, and the DNA coding rules are

used to scramble and diffuse the image, which creates a certain degree of security [20]. Wang et al. proposed to diffuse image pixels using five DNA operations (DNA addition, DNA subtraction, DNA XOR and DNA XNOR), and the Fisher–Yates scrambling scheme is introduced into the process of image scrambling to further enhance the encryption effect of the image [21]. Bhat et al. designed a color image encryption framework based on DNA encoding, the hyper-chaotic system and elliptic curve cryptography; the framework uses a hyper-chaotic system to dynamically control the coding rules of the image in each color channel, which enhances the randomness of the DNA sequence coding [22]. Wang et al. first used the improved zigzag transformation to scramble the image in the spatial domain and then encoded the image into DNA strands through DNA coding for further diffusion operation [23]. Zou et al. first encoded the image into a DNA strand and then performed the scrambling and diffusion operation completely on the DNA strand [15]. Its DNA strand operation mode is to construct long and short DNA strands for DNA strand exchange and diffusion processes, respectively. In general, most of the existing DNA coding encryption schemes have the problem of poor encryption efficiency, which needs to be further improved.

Aiming at further improving the efficiency of image encryption and ensuring the quality of encryption, we propose a DNA encoding image encryption scheme based on the 5D Hamiltonian conservative chaotic system. Firstly, SHA-256 is used to generate the initial value of the chaotic system and control the scrambling process of the DNA strand. Secondly, a new DNA strand generation method is proposed. Combined with DNA coding, the DNA strand is generated by combining the high-bit-plane and the low-bit-plane, and the effective information in the high-bit-plane is efficiently hidden, and the image information is initially hidden. In addition, the convolution sequence related to the plaintext is used to scramble the DNA strand to further scramble the original positions of the DNA strand bases. Finally, based on the operation rules of DNA operation, seven DNA index tables were constructed, and the diffusion process of the DNA strand was further controlled by retrieving the DNA index tables.

The main contributions of this paper are as follows:

1. A new method of DNA strand generation is explored, which takes DNA encoding as a bridge and combines the high-bit-planes with the low-bit-planes to hide the effective information in the high-bit-planes.
2. In order to further improve the encryption quality and ensure encryption efficiency, seven DNA operation index tables are constructed based on the operation rules of DNA computing. The diffusion operation of the DNA strand is carried out by searching the index table, which simplifies and accelerates the diffusion process of the DNA strand.

The remaining paper is structured as follows: Section 2 introduces some relevance theory; Section 3 explains the encryption and decryption process in detail; Section 4 shows the experimental results; Section 5 carries out the security analysis; Section 6 makes the conclusion and outlook.

2. Relevance Theory

2.1. 5D Harmiltonian Conservative Hyper-Chaotic System

High-dimensional chaotic systems exhibit fundamentally distinct characteristics compared to their low-dimensional counterparts, particularly through their expanded state space and intricate dynamical behaviors that significantly elevate state prediction complexity. The 5D Hamiltonian chaotic system belongs to the class of high-dimensional conservative hyper-chaotic systems, which exhibit high complexity and unpredictability. This system can evolve in a larger state space through nonlinear dynamics, avoiding the

periodicity or regularity that may arise in low-dimensional systems when initial conditions or parameters change. Therefore, this system provides stronger randomness for encryption applications and can effectively resist various types of attacks. Furthermore, the 5D Hamiltonian chaotic system preserves the energy conservation property under Hamiltonian dynamics conditions, which enables the system to maintain strong stability and sensitivity over long periods of evolution. By introducing higher dimensions, the system's degrees of freedom increase, making its dynamic behavior more complex, thereby enhancing the chaotic effects during the encryption process and further improving the security of image encryption [18].

The mathematical formulation of the 5D Hamiltonian conservative hyper-chaotic system is presented below:

$$\begin{cases} \dot{x} = xy + az \\ \dot{y} = -x^2 + z^2 + cw \\ \dot{z} = -yz + zw - ax + bv \\ \dot{w} = -z^2 + v^2 - cy \\ \dot{v} = -wv - bz \end{cases} \tag{1}$$

where $a, b, c = 10, 13, 10$ denote system parameters. This hyper-chaotic system manifests five Lyapunov exponents ($LE_1 = 1.18,\ LE_2 = 0.02,\ LE_3 = 0,\ LE_4 = -0.02,\ LE_5 = -1.18$), specifically comprising two positive exponents with the total sum of all five exponents equaling zero. This property mathematically verifies the system's Hamiltonian conservative nature, where energy preservation occurs despite its hyper-chaotic dynamics. The coexistence of pseudo-randomness and exceptional complexity, as demonstrated by the two positive Lyapunov exponents and five-dimensional phase space, establishes this system as a paradigm of conservative hyper-chaos. Visual evidence of its chaotic attractor structure is provided in Figure 1, showcasing the system's non-repeating yet energy-bound trajectory patterns.

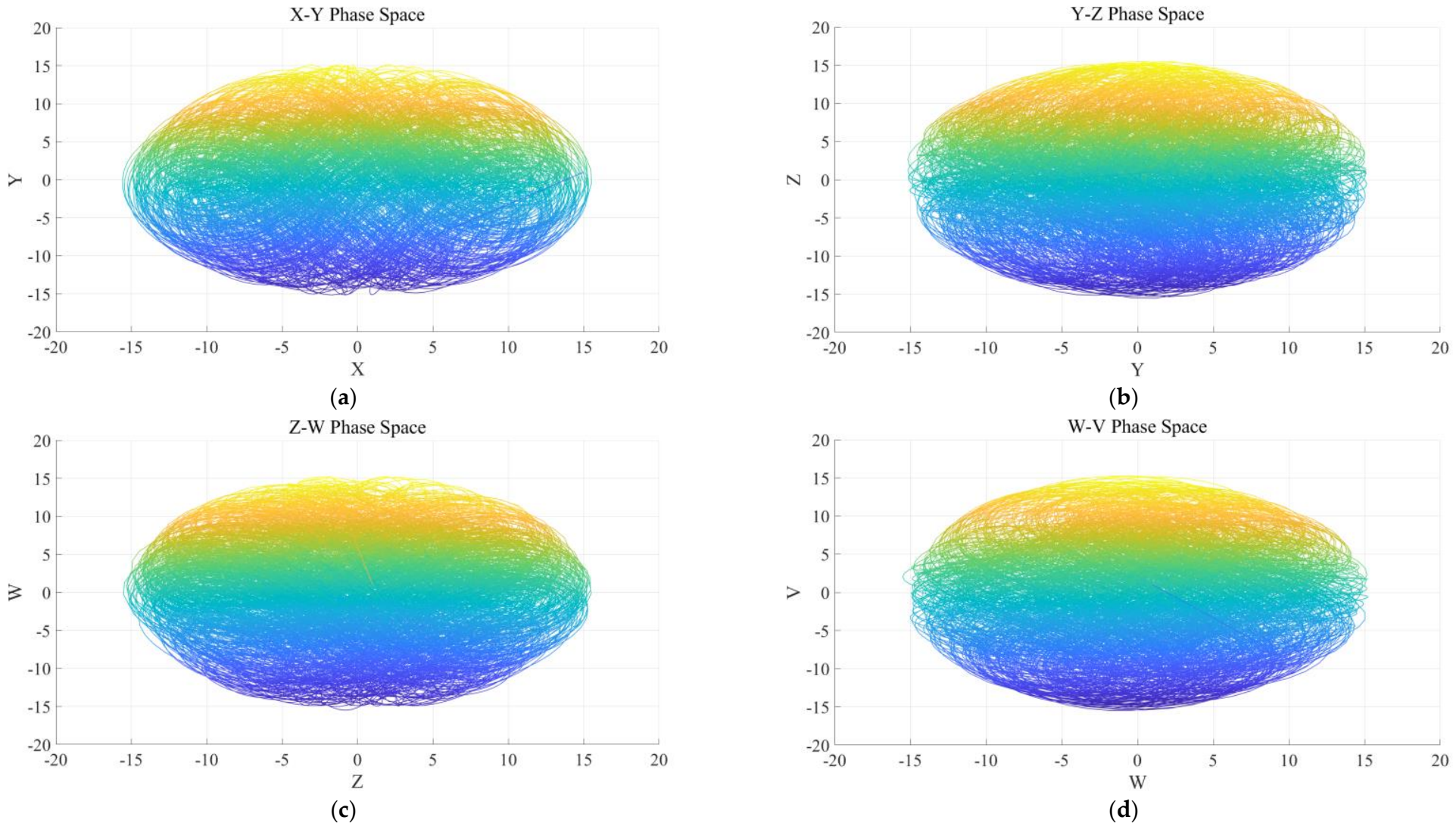


Figure 1. *Cont*.

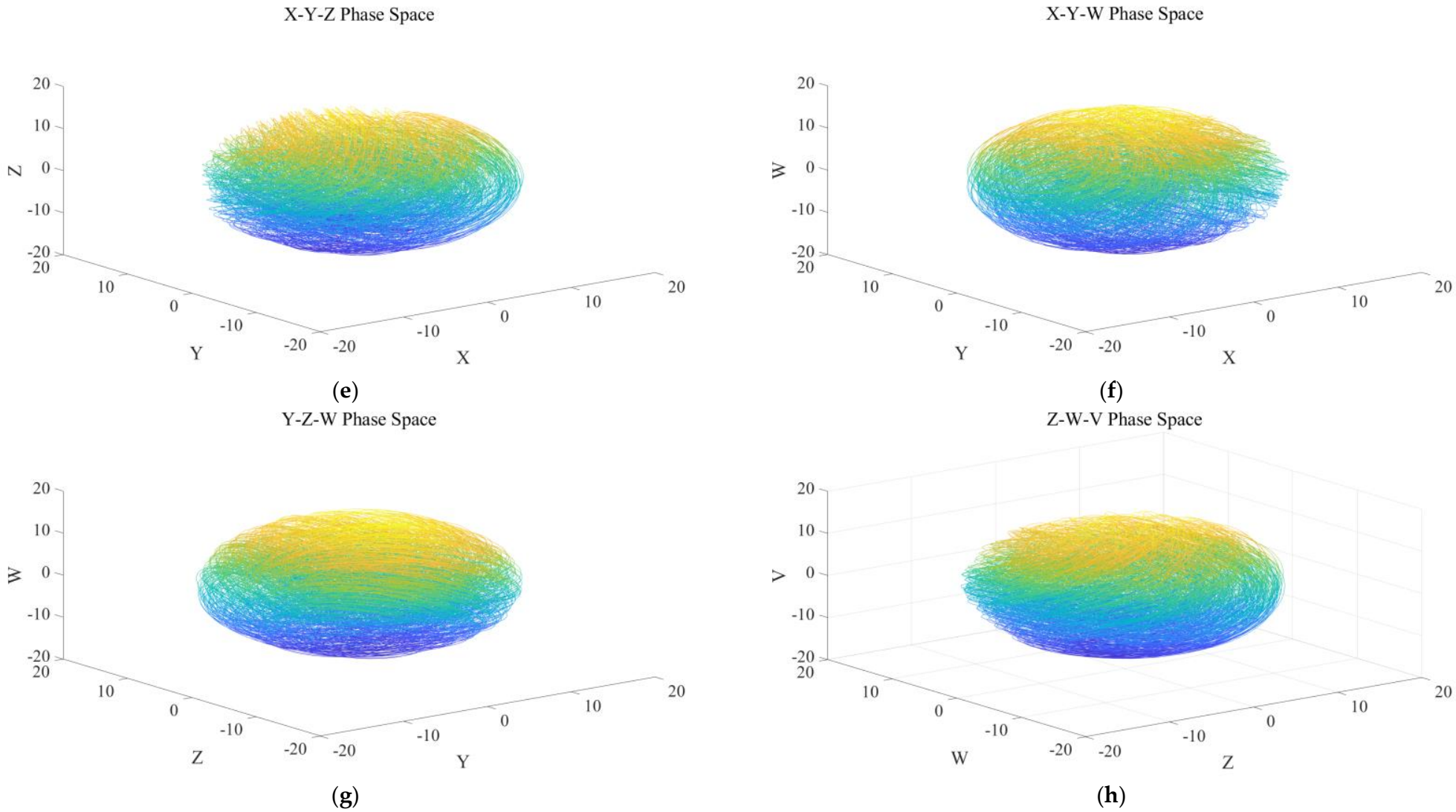


Figure 1. The chaotic attractor of 5D Harmiltonian conservative hyper-chaotic system: (**a**) X-Y phase space; (**b**) Y-Z phase space; (**c**) Z-W phase space; (**d**) W-V phase space; (**e**) X-Y-Z phase space; (**f**) X-Y-W phase space; (**g**) Y-Z-W phase space; (**h**) Z-W-V phase space.

2.2. Bit-Plane

An integer from 0 to 256 can be converted to an 8-bit binary number. For a grayscale image, the value of each pixel of the image is an integer in the range of 0 to 255. So, each gray value can be converted into an 8-bit binary number. 8 bits of the whole pixels can reconstruct 8 bits planes. As shown in Figure 2, From the 1st bit-plane to the 8th bit-plane, the clarity of the image is gradually improved, and the effective information in the image is gradually identifiable. The proportion of information carried by each bit-plane can be quantitatively calculated using the following formular.

$$p(k) = \frac{2^{k-1}}{\sum_{m=1}^{8} 2^{m-1}} \tag{2}$$

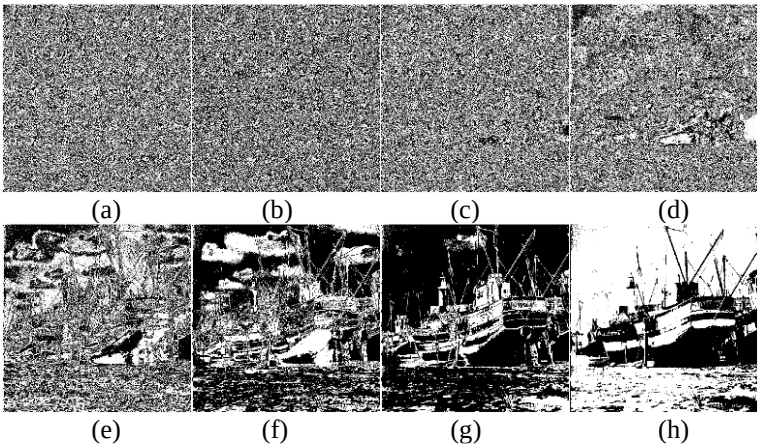


Figure 2. Bit-plane decomposition result of Image Boat. (**a**–**d**): low-bit-planes; (**e**–**h**): high-bit-planes.

As is listed in Table 1, the information-carrying ratio increases as the bit-plane rises. The information-carrying ratio of high-bit-planes (top 4) is almost 94%; on the contrary, the four low-bit-planes only account for nearly 5.8%. As a consequence, the four high-bit-planes contain most of the effective information in the image, and hiding the effective information in the high-bit-planes to the low-bit-planes can effectively ensure the security of the image. In this paper, a method based on DNA encoding is proposed to hide the effective information from the high-bit-planes to the low-bit-planes.

Table 1. The calculation result of proportion of information carried.

Bit-Planes	1th	2th	3th	4th	5th	6th	7th	8th
Proportion (%)	0.39	0.78	1.57	3.14	6.27	12.55	25.10	50.20
	5.88					94.12		

2.3. DNA Encoding and Decoding Rules

In the field of biology, deoxyribonucleic acid (DNA) is an important material for storing genetic information in biological cells. DNA consists of nucleotides with four different bases. The four bases are adenine (A), guanine (G), cytosine (C) and thymine (T). Based on the composition principle of four bases in DNA strand, two binary numbers are used to represent each base in cryptography. The four bases can form 24 combinatorial possibilities. However, according to Watson Crick's complement rule, A is paired with T and C is paired with G, only eight legal combinations exist, as shown in Table 2. In addition, There are many types of DNA arithmetic operations like XOR, XNOR, addition

and subtraction, right shift and left shift. The results of arithmetic operations are shown in Tables 3–6.

Table 2. DNA encoding and decoding rules.

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

Table 3. DNA arithmetic operations.

$\oplus/\otimes/+/-$	A	G	C	T
A	A/T/A/A	G/C/G/T	C/G/C/C	T/A/T/G
G	G/C/G/G	A/T/C/A	T/A/T/T	C/G/A/C
C	C/G/C/C	T/A/T/G	A/T/A/A	G/C/G/T
T	T/A/T/T	C/G/A/C	G/C/G/G	A/T/C/A

Table 4. Key space comparison.

	Proposed	[24]	[23]	[25]	[26]	[19]	[27]
Key space	2^{797}	2^{711}	2^{512}	2^{328}	2^{777}	2^{691}	2^{500}

Table 5. Difference between two cipher images.

Cipher Images		Key		Difference (%)
Figure 13a	Figure 13f	x_0	$x_0 + 10^{-16}$	99.46
Figure 13b	Figure 13g	y_0	$y_0 + 10^{-16}$	99.60
Figure 13c	Figure 13h	z_0	$z_0 + 10^{-16}$	99.62
Figure 13d	Figure 13i	w_0	$w_0 + 10^{-16}$	99.50
Figure 13e	Figure 13j	v_0	$v_0 + 10^{-16}$	99.67

Table 6. Variances of histograms.

Image	Clock	Bridge	Boat	Male
Plain	283,167.686	4,761,071.325	1,541,901.804	11,393,958.533
Cipher	332.463	902.086	953.765	3818.102

The security of DNA operation rules, such as XOR, XNOR, addition, subtraction, and shift operations, plays a crucial role in the effectiveness of the encryption algorithm. XOR operations are widely used due to their ability to produce significant randomness and diffusion, which are key for making the encrypted data unpredictable. However, XOR is susceptible to known-plaintext attacks, where attackers with partial information could reverse the operation and deduce the key. Similarly, the XNOR operation, being the complement of XOR, offers symmetry but also introduces potential vulnerabilities. Its predictable nature can be exploited if the key space is not sufficiently large, making it more prone to certain attacks.

Addition and shift operations (right and left shifts) contribute to the encryption by altering the position of the DNA sequence elements. These operations increase confusion and diffusion, preventing simple repetitive patterns. However, without sufficient key variation, they may produce predictable sequences that could be vulnerable to statistical attacks. Therefore, while these operations enhance encryption, their security depends heavily on the proper management of key space and the frequency of application.

In summary, while each DNA operation rule provides specific advantages in terms of increasing randomness, confusion, and diffusion, their security must be carefully balanced. The combination of multiple DNA operations with a large, unpredictable key space ensures robust resistance to cryptographic attacks, thus strengthening the overall encryption system.

3. Encryption and Decryption Process

3.1. Key Feature Extractor

Step 1: Suppose we have a sequence of decimal numbers, defined as key.

Step 2: Perform the bit XOR operation on the first and second values in the sequence Key; the result is denoted as *Keyfeature*.

Step 3: Perform the bit XOR operation on the value of *Keyfeature* and the next value of Key; the result is assigned to *Keyfeature*. By parity of reasoning, the value of *Keyfeature* is continuously subjected to bitwise XOR operation with the next value in the sequence Key until the last value in Key, the bitwise XOR operation is finished, and the final *Keyfeature* is obtained.

The algorithm of the key feature extractor is exhibited in Algorithm 1.

Algorithm 1: The Key Feature Extractor

Input : The sequence of decimal numbers Key
Output : The value of $Keyfeature$
$Keyfeature = bitxor(Key(1),\ Key(2))$;
$n = size(Key, 2)$;
$for\ i = 3 : n$
  $Keyfeature = bitxor(Keyfeature, Key(i))$;
end
$return\ Keyfeature$

3.2. Plaintext-Related Sequence Generation

Step 1: Use SHA-256 to calculate the hash string of plain image in three-direction.

Step 2: Convert the previous hash string to 32 bits decimal sequence s and sort them in ascending order.

Step 3: Perform feature extraction on the sorted decimal sequence s by key feature extractor to obtain $keyfeature1$. Then, divide s into four sub-sequences. The four sub-sequence lengths are 5, 9, 9, and 9, respectively. Each sub-sequence in turn takes the sequence of corresponding length in the sequence s.

Step 4: The first sub-sequence is described as Key. Then, reshape the second, third and fourth sub-sequences to three 3×3 sequences (S_conv1, S_conv2, S_conv3). S_conv4 is obtained by mod operation on S_conv3:

$$S_conv4 = \mathrm{mod}(S_conv3, keyfeature1) \tag{3}$$

From Step 1 to Step 4 can be summarized as the process of generating the plaintext-related sequences, as shown in Figure 3.

Step 5: The four convolutional sequences S_conv1', S_conv2', S_conv3' and S_conv4' are combined and then the combined sequence is reshaped as a row vector S_conv of size $M \times N \times 4$:

$$\begin{cases} S_conv = [S_conv1; S_conv2; S_conv3; S_conv4] \\ S_conv = reshape(S_conv, 1, 4 \times M \times N) \end{cases} \tag{4}$$

The obtained row vector S_conv is used as the permutation sequence of the encryption and decryption process.

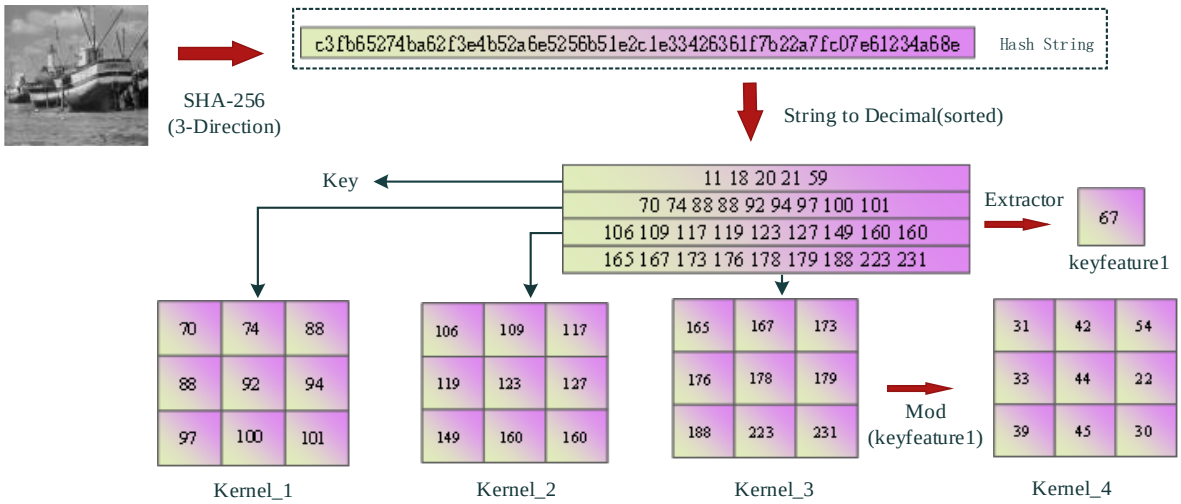


Figure 3. Plaintext-related sequence generator.

3.3. Key Stream Generator

Step 1: The 5D Harmiltonian conservative hyper-chaotic system needs five initial values to generate five pseudo-random sequences. The calculation formular is given below:

$$\begin{cases} x_0 = \alpha \sin\phi + \cos\{\sum_{i=1}^{n}[p(i)/\pi \times 10^{-6}]\} \\ y_0 = \beta \sin\psi + \cos\{\sum_{i=1}^{n}[S_conv1(i)/\pi \times 10^{-9}]\} \\ z_0 = \gamma \sin\varphi + \cos\{\sum_{i=1}^{n}[S_conv2(i)/\pi \times 10^{-10}]\} \\ w_0 = \sigma \sin\chi + \cos\{\sum_{i=1}^{n}[S_conv3(i)/\pi \times 10^{-10}]\} \\ v_0 = \delta \sin\theta + \cos\{\sum_{i=1}^{n}[S_conv4(i)/\pi \times 10^{-10}]\} \end{cases} \quad (5)$$

where n denotes the size of plain image. $\alpha, \beta, \gamma, \sigma, \delta, \phi, \psi, \varphi, \chi, \theta$ denote the control parameter of the initial sequence. These ten parameters are not randomly chosen but depend on the plaintext image and multiple features extracted by the key feature extractor. The ten parameters are set as follows:

$$\begin{cases} \alpha = key(1), \beta = key(2), \gamma = key(3), \sigma = key(4), \delta = key(5) \\ \phi = keyfeature1, \psi = keyfeature2, \varphi = keyfeature3, \chi = keyfeature4, \theta = keyfeature5 \end{cases} \quad (6)$$

Step 2: Substituting Equation (6) into Equation (5) results in five initial values x_0, y_0, z_0, w_0, v_0 of the hyper-chaotic system. According to Equation (1), substituting x_0, y_0, z_0, w_0, v_0 into Equation (1) obtains a pseudo-random matrix PR with $M \times N$ rows and 5 columns.

Step 3: Perform three different transformation operations on PR, as shown below:

$$\begin{cases} PR_1 = \mathrm{mod}(PR \times 2^{16}, 256) \\ PR_2 = \mathrm{mod}((PR + 100) \times 10^{10}, M \times N) + 1 \\ PR_3 = \mathrm{mod}((PR + 100) \times 10^{14}, 256) + 1 \end{cases} \quad (7)$$

Step 4: Use PR_1, PR_2, PR_3 to further generate the key stream needed in DNA strand encoding. DNA strand confusion and diffusion are shown below:

$$\begin{cases} KM = reshape(PR_1(:,1), M, N) \\ KE = reshape(PR_1(:,2:end), 1, M \times N \times 4) \\ KP = reshape(PR_3(:,2:end), 1, M \times N \times 4) \\ KX = setdiff(1:M \times N, unique(PR_2(:,2:end))) \end{cases} \tag{8}$$

The flowchart of the whole DNA strand generation is shown in Figure 4. The specific steps are as follows:

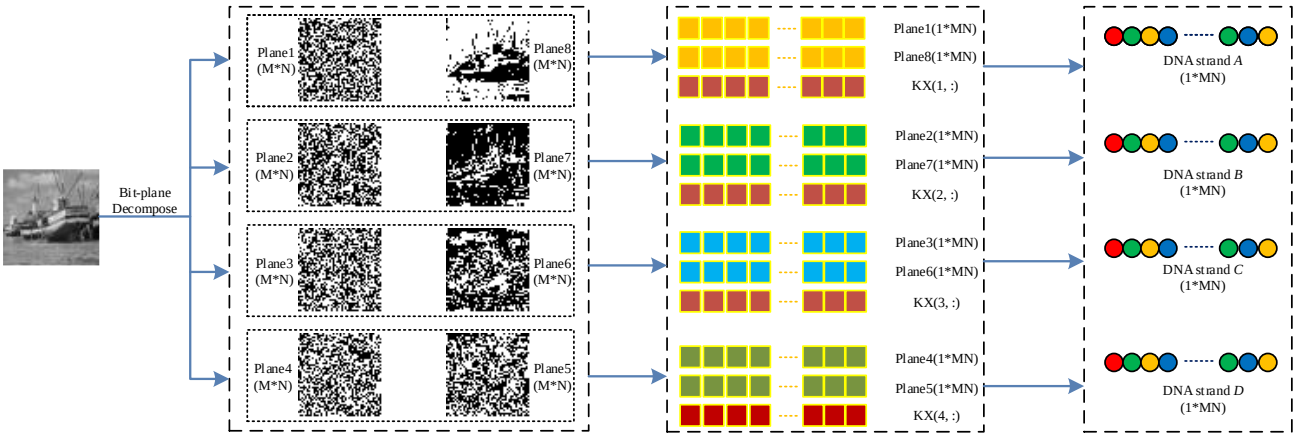


Figure 4. The flowchart of DNA strand generation.

Where $unique()$ means removing duplicate elements from the sequence, $setdiff(A, B)$ denotes the elements not present, and A is inserted into the end of the sequence B. KM is used as a mask image of the process of image encryption. KE is the indicator sequence of DNA strand encoding. KP is the indicator sequence of DNA strand diffusion. KX is the indicator sequence of DNA strand generation.

3.4. DNA Strand Generation

The flowchart of the whole DNA strand generation is shown in Figure 4. The specific steps are as follows:

Step 1: Perform bit-plane decomposition on the plain image to obtain eight bit-planes (plane i, $i \in [1, 8]$). The high-bit-plane contains more effective information of the image than the low-bit-plane. Therefore, the high-bit-plane can be fused with the low-bit-plane to generate a DNA strand, and the effective information in the image can be hidden in the DNA strand as much as possible.

Step 2: Divide the eight bit-planes into four groups; plane and plane $9 - i$ are divided into one group, $i \in [1, 8]$.

Step 3: Transform two bit-planes of each group to two row-vectors of length MN, then use key stream KX to control the fusion process of the transformed high-bit-plane and low-bit-plane. Each group of bit-planes can generate one DNA strand. Therefore, four DNA strands A, B, C, D are generated. The DNA strand generation process is shown in Figure 5.

As shown in Figure 5, $KX(i,:)$ is the index sequence of Plane $9 - i$; Plane i and Plane $9 - i$ are the corresponding row-vectors of low-bit-planes and high-bit-planes; binary sequence denotes the sequence after the fusion of Plane KX and Plane i; the DNA strand denotes the final generation DNA sequence. The following is illustrated with the first element in the sequence KX:

The 1st value of KX is 3. Find that the 3rd element of the Plane $9 - i$ is 0, merge with the 1st element of the sequence in Plane i, obtain the 1st binary value 10 in the binary

sequence, then according to the first encoding rule of DNA encoding in Table 2, obtain the 1st element in the DNA strand. Similarly, by traversing all the elements in the keystream KX, the two-bit-planes can be combined to generate the corresponding DNA strand.

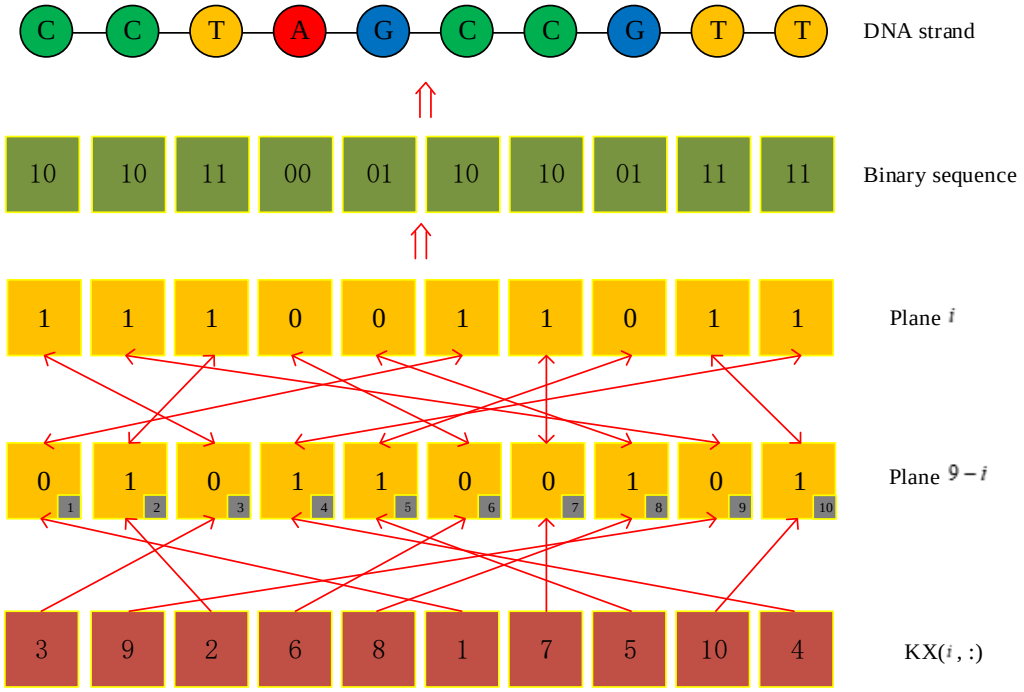


Figure 5. An example of DNA strand generation process.

3.5. DNA Strand Diffusion

To completely ensure the randomness of each base in the DNA strand, we adopt eight DNA encoding rules instead of a single one to encode the DNA strands, which greatly improves the security of the DNA strand encoding results.

Step 1: Splice the four DNA strands (A, B, C, D) into a single DNA strand with a length of $4 \times MN$.

Step 2: Since each value in the key stream KE is an integer between 0 and 255, KE cannot be directly used to control the selection of encoding rules. It is necessary to transform the value range of the keystream KE to be between 1 and 8. The transformation formula is given below:

$$\begin{cases} [\sim, q] = sort(S_conv) \\ ps = eds(q) \end{cases} \tag{9}$$

Step 3: R is used to control the process of DNA strand encoding. Each element in the sequence R indicates the encoding rule used by the base at the corresponding position in the DNA strand. As is shown in Figure 6, the first element of the sequence R is 7, so the first base of DNA strand (C) uses Rule 7 to encode the base C, and the base C has transformed the base T under the influence of Rule 7. Similarly, the other bases in the DNA strand are encoded in the same way to obtain final encoded DNA strand eds.

Step 4: As shown in Equation (4), use S_conv permutation sequence to permute the DNA strand.

$$\begin{cases} [\sim, q] = sort(S_conv) \\ ps = eds(q) \end{cases} \tag{10}$$

where q denotes the index of permutation, eds represents the encoded DNA strand, and ps denotes the permutated DNA strand.

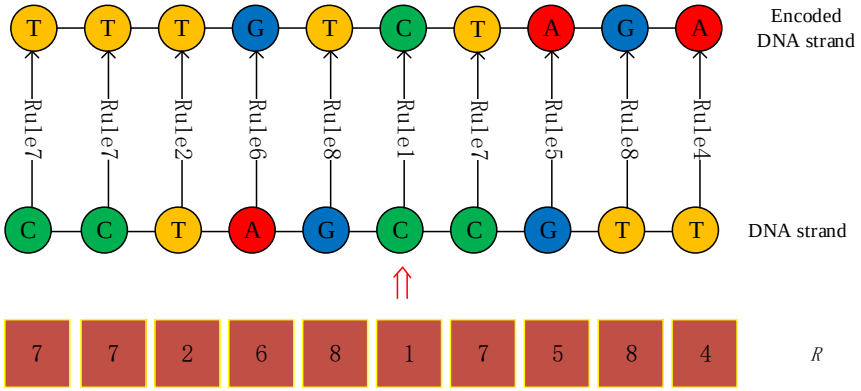


Figure 6. An example of DNA strand permutation process.

3.6. DNA Strand Diffusion

In this stage, seven operations are used to perform DNA strand diffusion (XOR, Addition, Multiple, XNOR, Subtract, Right-Shift, Left-Shift).

Step 1: Since each value in the key stream KP is an integer between 0 and 255, KP cannot be directly used to control the selection of operation rules. It is necessary to transform the value range of the keystream KP to be between 1 and 7. The transformation formula is given below:

$$op = floor(\mathrm{mod}(KP, 7)) + 1 \tag{11}$$

Step 2: Construct the operation table, as is shown in Equation (12):

$$\begin{gathered} XOR = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix}, ADD = \begin{bmatrix} 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \\ 2 & 3 & 0 & 1 \end{bmatrix}, MUL = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 2 & 3 & 0 & 1 \\ 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}, XNOR = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 2 & 3 & 0 & 1 \\ 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}, \\ SUB = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 3 & 0 & 1 \end{bmatrix}, RShift = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}, LShift = \begin{bmatrix} 0 & 3 & 2 & 1 \\ 1 & 0 & 3 & 2 \\ 2 & 1 & 0 & 3 \\ 3 & 2 & 1 & 0 \end{bmatrix}. \end{gathered} \tag{12}$$

Step 3: The seven operations all need two DNA strands. Only one DNA strand cannot meet the need of the process of DNA strand diffusion. KM is used as a mask image of the process of image encryption. Therefore, the KM is used to generate another DNA strand. Similarly, the bit-plane decomposition is performed first, and the corresponding DNA strand is generated. Through the above process, the mask image KM is transformed into another DNA strand pt.

Step 4: The key stream op is used to control the process of DNA strand encoding. Each element in the sequence op indicates the operation rule used by the base at the corresponding position between the encoded DNA strand ps and the mask DNA strand pt.

Step 5: Since the operation tables are defined in Equation (12), the diffused DNA strand pd can be obtained only by looking up the corresponding table according to the control sequence op. The table lookup method can be defined as follows:

$$f_{operate}(base1, base2) = operate(base1 + 1, base2 + 1) \tag{13}$$

where $f_{operate}(.)$ represents $f_{XOR}(.)$, $f_{ADD}(.)$, $f_{MUL}(.)$, $f_{XNOR}(.)$, $f_{SUB}(.)$, $f_{RShift}(.)$, and $f_{LShift}(.)$; $operation(.)$ denotes the corresponding operation table, as shown in Equation (7). This denotes the base of the sequence ps, and the $base2$ denotes the base of the sequence pt. The values of the four bases A, G, C, T are 0, 1, 2, 3. For example, base1 = A, base2 = T, $f_{XOR}(A, T) = XOR(A + 1, T + 1) = XOR(1, 4) = 3 = T$.

As is shown in Figure 7, the first element of the sequence op is 3, so the third operation table MUL table is used; the first elements of ps and pt are base C and base T. According to the Equation (13), the first element of diffused DNA strand pd can be obtained:

$$f_{MUL}(C,T) = MUL(C+1, T+1) = MUL(3,4) = 2 = C \tag{14}$$

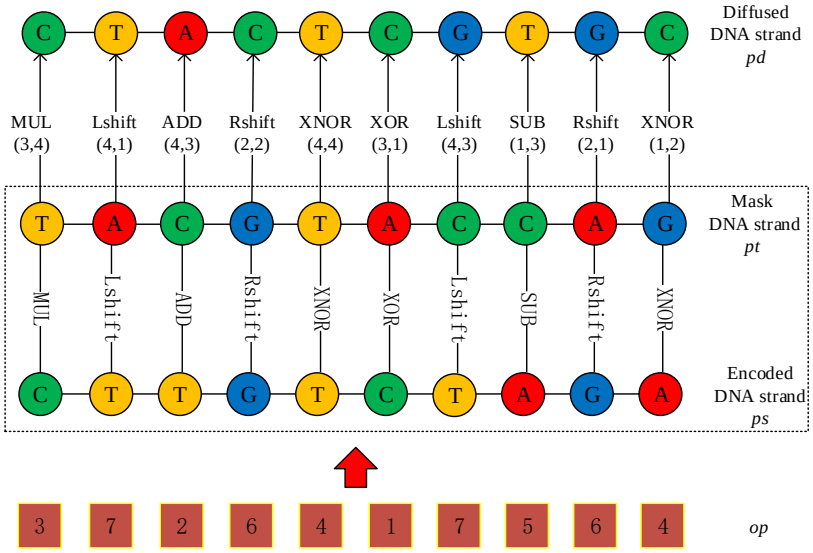


Figure 7. An example of DNA strand diffusion process.

Consequently, the first element of diffused DNA strand pd is base C. The algorithm of DNA strand diffusion is shown in Algorithm 2.

Step 6: Use the DNA encoding rules, which are shown in Table 2, to decode the diffused DNA strand pd to get eight bit-planes, then combine the eight bit-planes to generate the final encrypted image.

The flowchart of the image encryption process is shown in Figure 8.

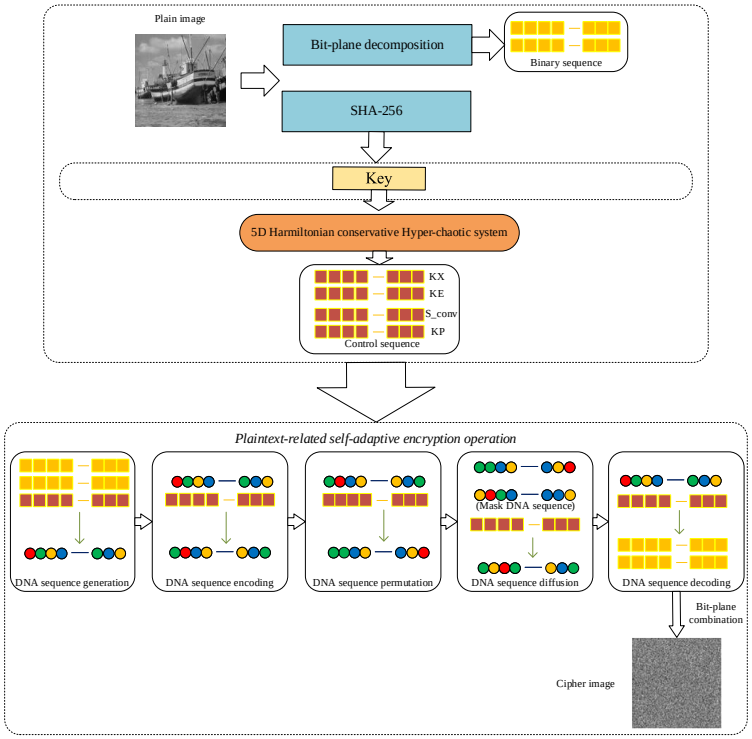


Figure 8. Image encryption process.

Algorithm 2: The Process of DNA Strand Diffusion

Input : Masked DNA strand pt, Encoded DNA strand ps, the control sequence op
Output : The diffused DNA strand pd

// the length of both DNA strands and control sequence is $4 * m * n$
$XOR = [0\ 1\ 2\ 3;\ 1\ 0\ 3\ 2;\ 2\ 3\ 0\ 1;\ 3\ 2\ 1\ 0]$;
$ADD = [1\ 0\ 3\ 2;\ 0\ 1\ 2\ 3;\ 3\ 2\ 1\ 0;\ 2\ 3\ 0\ 1]$;
$MUL = [3\ 2\ 1\ 0;\ 2\ 3\ 0\ 1;\ 1\ 0\ 3\ 2;\ 0\ 1\ 2\ 3]$;
$XNOR = [3\ 2\ 1\ 0;\ 2\ 3\ 0\ 1;\ 1\ 0\ 3\ 2;\ 0\ 1\ 2\ 3]$;
$SUB = [1\ 2\ 3\ 0;\ 0\ 1\ 2\ 3;\ 3\ 0\ 1\ 2;\ 2\ 3\ 0\ 1]$;
$RShift = [0\ 1\ 2\ 3;\ 1\ 2\ 3\ 0;2\ 3\ 0\ 1;\ 3\ 0\ 1\ 2]$;
$LShift = [0\ 3\ 2\ 1;\ 1\ 0\ 3\ 2;2\ 1\ 0\ 3;\ 3\ 2\ 1\ 0]$;
$for\ i = 1 : 4 * m * n$
  $switch\ op(i)$
    $case\ 1$
      $pd(i) = ADD(ps(i) + 1, pt(i) + 1)$;
    $case\ 2$
      $pd(i) = SUB(ps(i) + 1, pt(i) + 1)$;
    $case\ 3$
      $pd(i) = XOR(ps(i) + 1, pt(i) + 1)$;
    $case\ 4$
      $pd(i) = XNOR(ps(i) + 1, pt(i) + 1)$;
    $case\ 5$
      $pd(i) = MUL(ps(i) + 1, pt(i) + 1)$;
    $case\ 6$
      $pd(i) = RShift(ps(i) + 1, pt(i) + 1)$;
    $case\ 7$
      $pd(i) = LShift(ps(i) + 1, pt(i) + 1)$;
  end
end
$return\ pd$

3.7. Decryption Process

In general, the decryption process is the reverse process of the image encryption process. The flowchart is shown in Figure 9. The specifics are described below:

Step 1: Substitute Key and *keyfeature* into Equations (5) and (6) to obtain the initial value of the chaotic system. Then, use the 5D Harmiltonian conservation hyper-chaotic system to generate four key streams KE, KP, KM, KX.

Step 2: Perform bit-plane decomposition operation on cipher image and mask image KM.

Step 3: Transform every bit-planes of cipher image to row vectors of length MN, the two vectors can generate one DNA strand, so the eight row vectors can generate four DNA strands, then splice four DNA strands with the length of M*N to obtain the required length of the 4*M*N DNA strand.

Step 4: Perform bit-plane decomposition operation on mask image, then perform DNA strand generation to obtain another DNA strand PT with the length of 4*M*N.

Step 5: Use key stream KE to control DNA strand encoding, then we can obtain the encoded DNA strand.

Step 6: Remove the diffusion state of the encoded DNA strand by another DNA strand PT and the diffusion control sequence KP.

Step 7: Remove the permutation state of the encoded DNA strand by sequence S_conv.

Step 8: Use *KE* to perform DNA strand decoding, then use *KX* to recover the bit-plane. Finally, merge the bit-planes to obtain the final encrypted image.

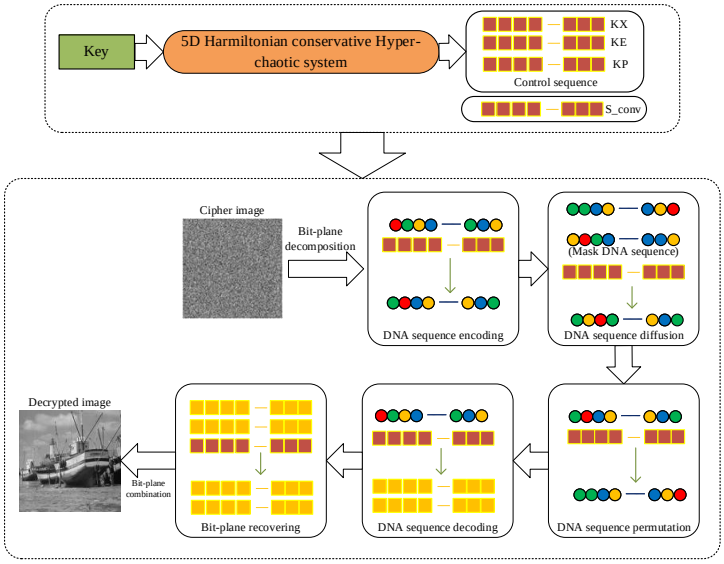


Figure 9. Image decryption process.

4. Experiment Results

To demonstrate the effectiveness of our algorithm, the experiments are divided into two groups, grayscale image encryption experiment and color image encryption experiment. As is shown in Figure 10, Group 1 contains four grayscale images (Clock, Bridge, Boat, Male). As is shown in Figure 11, Group 2 contains four color images (Female, Pine, House, Fighter). The experiment result shows that our encryption algorithm is effective for grayscale image and color image, the cipher image of both cannot be identified at the same time.

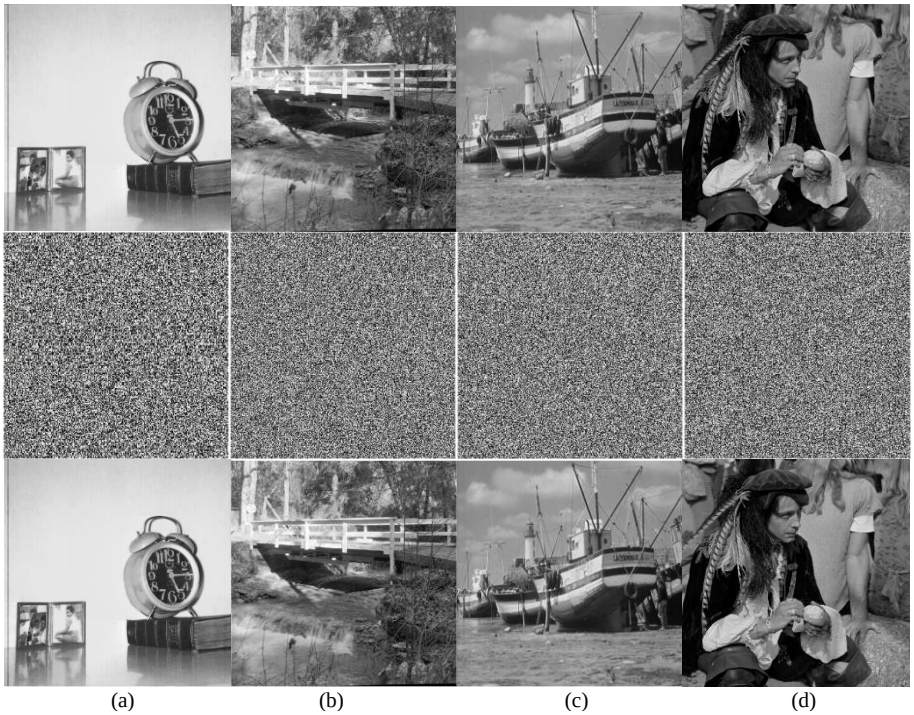


Figure 10. Encryption results of Group 1: (**a**) clock (256 × 256); (**b**) bridge (512 × 512); (**c**) boat (512 × 512); (**d**) male (1024 × 1024). First line: plain images; second line: encrypted images; third line: decrypted images.



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

Figure 11. Encryption results of Group 2: (**a**) female (256 × 256 × 3); (**b**) pine (256 × 256 × 3); (**c**) house (512 × 512 × 3); (**d**) fighter (512 × 512 × 3). First line: plain images; second line: encrypted images; third line: decrypted images.

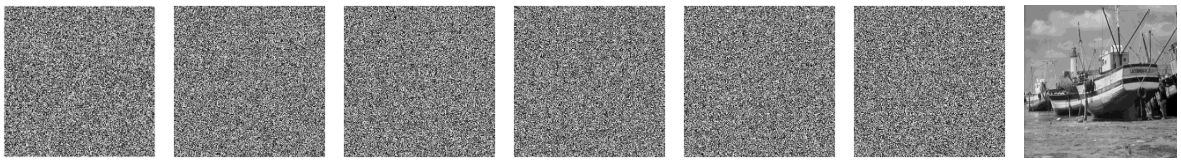
5. Security Analyses

5.1. Key Space Analysis

As is shown in Equation (5), the Keys are as follows: $x_0, y_0, z_0, w_0, v_0, \alpha, \beta, \gamma, \sigma, \delta, \phi, \psi, \varphi, \chi, \theta$. The precision of computers is usually 64 bits. Therefore, the key space is $\left(10^{16}\right)^{15} = 10^{240} \approx 2^{797} > 2^{100}$. This proves that the key space of our method is large enough to defend against exhaustive attacks. Table 4 shows the result of key space comparison. The comparison results prove that the key space of our algorithm has better performance than other similar image encryption methods using DNA encoding.

5.2. Key Sensitivity Analysis

An effective encryption algorithm should ensure that even if the attacker fine-tunes any value in the key, it can still ensure that the modified key cannot be decrypted to get the original image. As is shown in Figure 12, The decrypted image cannot be recognized with a very small fine-tuning of any value in the key (even 10^{-16}), and the original image can only be decrypted with the original correct key.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**)

Figure 12. Key sensitivity experiment: (**a**) cipher image "Boat"; (**b**–**f**) use incorrect key to decrypt: $x_0 + 10^{-16}$, $y_0 + 10^{-16}$, $z_0 + 10^{-16}$, $w_0 + 10^{-16}$, $v_0 + 10^{-16}$; (**g**) use correct key to decrypt.

We also investigate the effect of fine-tuning the key on the cipher image. The differences between the original cipher image and fine-turned key cipher images are shown in Figure 13. The quantified difference rates are shown in Tables 7 and 8, and the difference

rate is up to 99.67%. The experiment results show that making even small changes to the secret key can have a dramatic impact on the cipher image; there is almost no similarity between the original ciphertext image and the fine-tuned key ciphertext image.

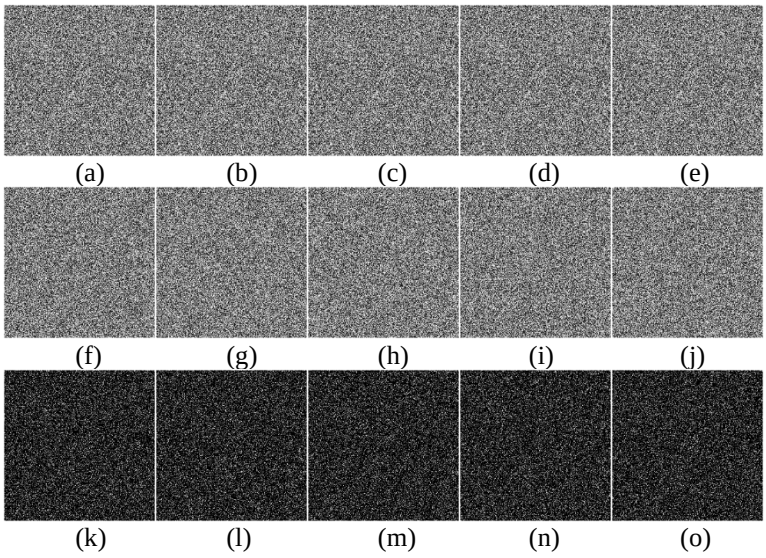


Figure 13. Cipher image difference ("Boat"). (**a**–**e**): cipher image using x_0, y_0, z_0, w_0, v_0; (**f**–**j**): cipher image using $x_0 + 10^{-16}$, $y_0 + 10^{-16}$, $z_0 + 10^{-16}$, $w_0 + 10^{-16}$, $v_0 + 10^{-16}$; (**k**–**o**): difference between cipher images (**a**–**e**) and cipher images (**f**–**j**).

Table 7. Variances of histograms with different keys.

Cipher Image	Key	α	β	γ	σ	δ	ϕ	ψ	φ	χ	θ
Bridge	902.086	885.467	974.031	986.126	887.224	850.361	965.992	928.494	869.106	940.251	961.851
Boat	953.765	1026.784	874.314	968.047	975.937	1022.620	982.737	973.702	1024.275	908.267	1027.388
Couple	1036.478	1004.761	945.247	1001.655	1088.980	1031.922	994.800	1078.549	1136.659	1055.271	1008.400
Tank	1003.639	1043.333	1083.906	1110.706	1044.965	988.329	1053.325	1009.741	1034.439	928.816	1035.663
Average	973.992	990.086	969.375	1016.634	999.276	973.308	999.214	997.622	1016.120	958.151	1008.325

Table 8. Variance change rate of different keys.

Cipher Image	α	β	γ	σ	δ	ϕ	ψ	φ	χ	θ
Bridge	1.84%	7.98%	9.32%	1.65%	5.73%	7.08%	2.93%	3.66%	4.23%	6.63%
Boat	7.66%	8.33%	1.50%	2.32%	7.22%	3.04%	2.09%	7.39%	4.77%	7.72%
Couple	3.06%	8.80%	3.36%	5.07%	0.44%	4.02%	4.06%	9.67%	1.81%	2.71%
Tank	3.96%	8.00%	10.67%	4.12%	1.53%	4.95%	0.61%	3.07%	7.46%	3.19%
Average	1.65%	0.47%	4.38%	2.60%	0.07%	2.59%	2.43%	4.33%	1.63%	3.53%

5.3. Statistic Analyses

5.3.1. Histogram Analysis

Figure 14 shows the 3D histograms of different grayscale images and color images. It can be clearly observed from the figure that the histograms of plain images are not uniform; on the contrary, the histogram of the encrypted image is uniform. At the same time, the variance of the histogram is also used to quantitatively analyze the degree of uniformity of the histogram. A larger variance indicates a poorer uniformity of the image, and more information is contained in the image. The calculation formula is as follows:

$$\mathrm{var}(w) = \frac{1}{l^2}\sum_{ij}\frac{1}{2}(w_i - w_j)^2 \tag{15}$$

where w_i and w_j represent the cumulative number of pixels in the image whose pixel values are equal to i and j. l = 256 denotes the 256 grayscale levels in the histogram. Table 9 lists the variances of histograms of the four grayscale images shown in Figure 14a,e,i,m. It is observed that the variance of the histogram of the encrypted image is far less than that of the original image. Table 10 shows strands for the histogram variance of different grayscale images under different keys. The first column variance of Table 10 corresponds to the unchanged key, and the variance of the other columns is obtained by changing the $\alpha, \beta, \gamma, \sigma, \delta, \phi, \psi, \varphi, \chi, \theta$ in the key. Table 11 notes the change rate of each changed key compared with the original key. It can be seen from Table 11 that only when the change rate of variance reaches 10.67% is it at its highest. In addition, all the average change rates are below 4.5%. The variance change rate of different keys demonstrates that the performance of the histogram is entirely dependent on the plain image. Table 12 indicates the comparison result of histogram variances, and the proposed algorithm has the lowest variance compared to the other similar algorithms. The experiment result indicates that the proposed algorithm can resist any kind of statistical attack.

Table 9. Comparison of histogram variances.

Methods	Lena	Barbara	Peppers	Male
Proposed	982.808	986.345	959.671	3818.102
[24]	993.570	1042.672	967.367	4020.578
[28]	1047.655	941.168	-	-
[29]	5335.83	5013.88	-	-

Table 10. Correlation coefficients of different images.

Images	Plain Image			Cipher Image		
	H	V	D	H	V	D
Clock	0.95632	0.97219	0.93790	0.00383	−0.00007	0.00606
Bridge	0.94046	0.91962	0.89272	−0.00496	−0.00299	0.00538
Boat	0.93446	0.97153	0.92095	0.00660	−0.00848	−0.00036
Male	0.97944	0.97910	0.96400	0.00154	−0.00418	0.00716
Couple	0.93419	0.88103	0.84819	0.00762	−0.00273	0.00162
Tank	0.94344	0.93047	0.89739	0.00304	−0.00660	0.00528
Lena	0.96774	0.98533	0.95554	−0.00109	−0.00039	0.00007
Peppers	0.97200	0.97781	0.95525	0.00027	−0.00042	−0.00076

Table 11. Correlation coefficients comparison of Peppers image.

Algorithms	Cipher Image		
	H	V	D
Proposed	0.00027	−0.00042	−0.00076
[24]	−0.0002	−0.0029	−0.0061
[21]	0.0089	−0.0037	−0.0042
[30]	0.0139	0.0054	0.0153
[31]	−0.0013	0.0019	0.0009

Table 12. Correlation coefficients comparison of Lena image.

Algorithms	Cipher Image		
	H	V	D
Proposed	−0.00109	−0.00039	0.00007
[32]	−0.0020	−0.0065	0.0087
[33]	−0.0017	−0.0009	−0.0019
[34]	−0.0239	−0.0033	0.0046
[35]	0.0115	0.0193	0.0234
[36]	0.0085	0.0054	0.0049
[15]	0.0004	−0.0033	−0.0070

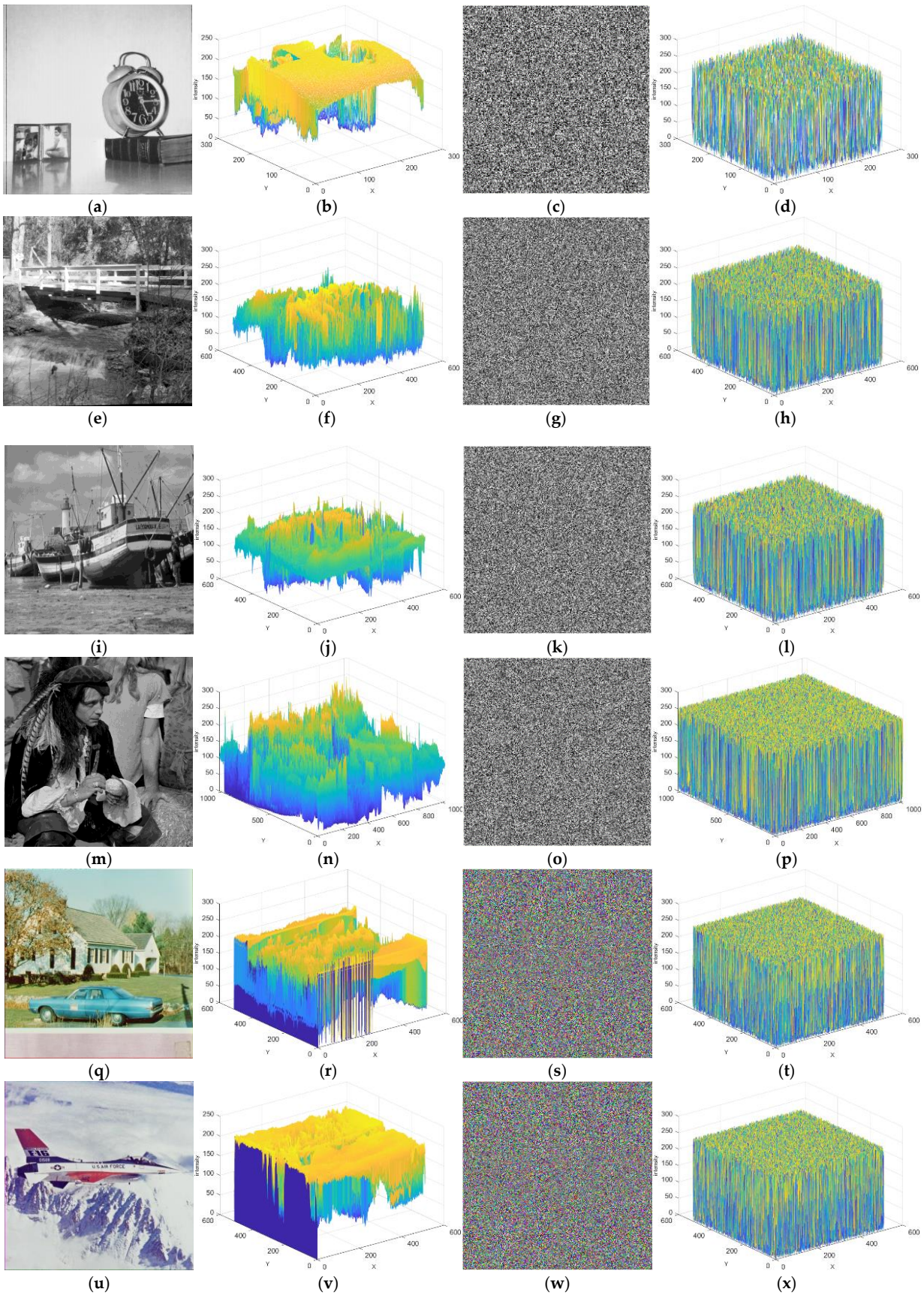


Figure 14. Image 3D histogram. (**a**,**e**,**i**,**m**,**q**,**u**): plain images; (**b**,**f**,**j**,**n**,**r**,**v**): 3D histograms of plain images; (**c**,**g**,**k**,**o**,**s**,**w**): cipher images; (**d**,**h**,**l**,**p**,**t**,**x**): 3D histograms of cipher images.

5.3.2. Correlation Analysis of Adjacent Pixels

The adjacent pixel correlation is usually used to describe the correlation degree of pixel values in adjacent image positions, which contain horizontal (H), vertical (V), and diagonal (D) three directions. An excellent image encryption algorithm can effectively decrease the correlation coefficient and make it extremely close to 0. Figure 15 shows the adjacent pixel correlation distribution of plain images and corresponding cipher images. As is shown in Figure 15, the correlation distribution of plain images is concentrated near the diagonal of the histogram, and the adjacent values of the corresponding cipher image are uniformly distributed in the histogram. Table 10 lists the correlation coefficient of different plain images and corresponding cipher images. Table 11 illustrates the comparison results of different algorithms with respect to the correlation coefficient of the Pepper image. Table 12 compares the performance of the correlation coefficient of Lena image with different algorithms. From the comparison results of Figures 16 and 17, it can be concluded that the correlation coefficient of the proposed algorithm has better performance than that of other similar algorithms. As a consequence, the proposed algorithm can effectively prevent statistical attacks.

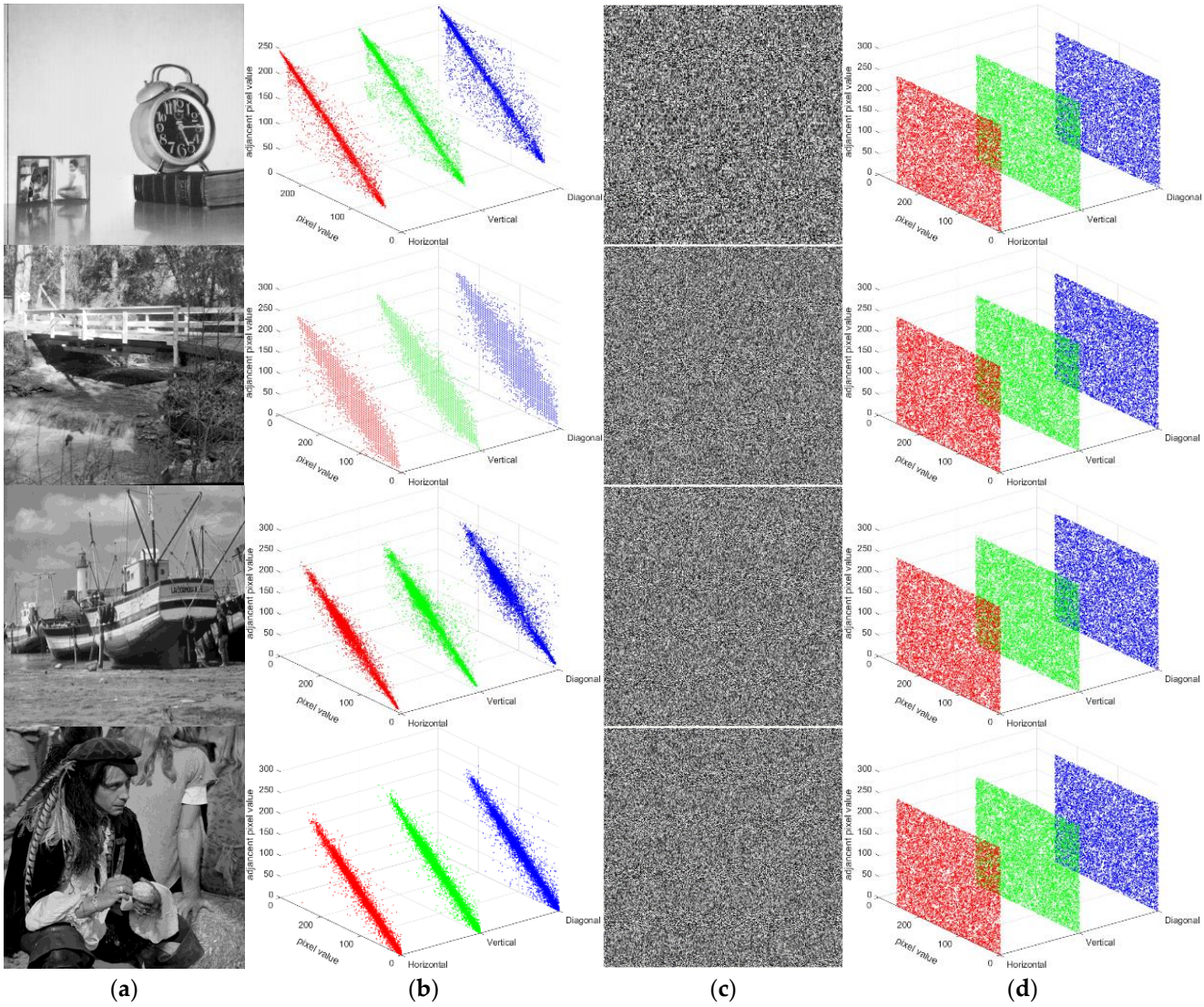


Figure 15. Adjacent pixel correlation distribution. (**a**): plain images; (**b**): correlation distribution of plain images; (**c**): corresponding cipher images; (**d**): correlation distribution of cipher images.

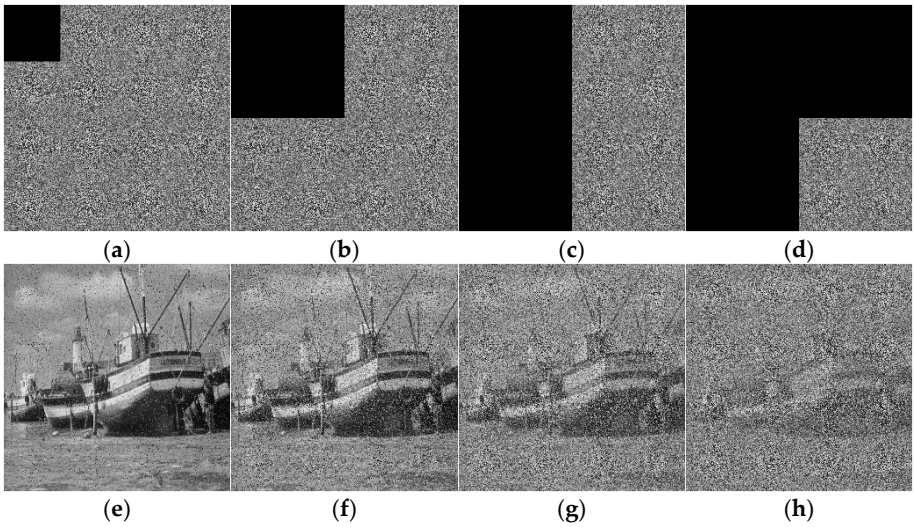


Figure 16. The results of the cropping attacks: (**a**–**d**) cipher image of 1/16, 1/4, 1/2, 3/4 cropping attacks; (**e**–**h**) corresponding decrypted image of different cropping attacks.

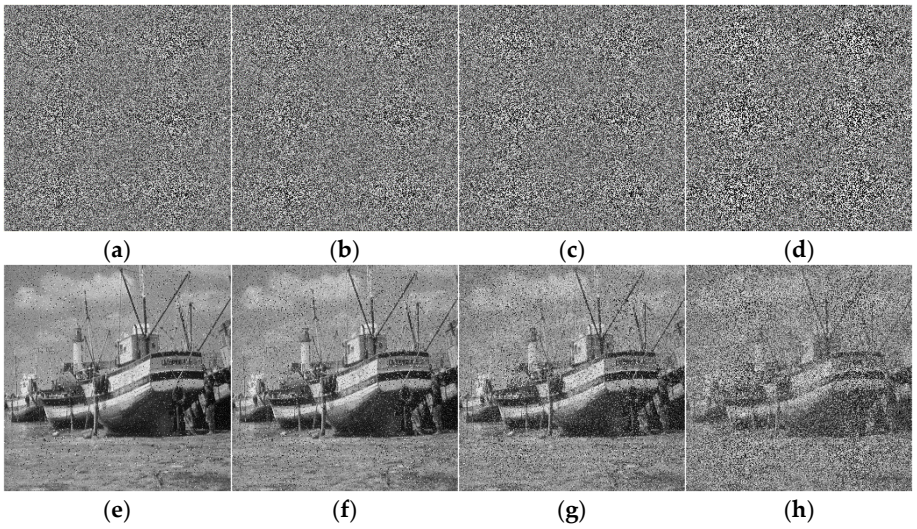


Figure 17. The results of the noise attacks: (**a**–**d**) cipher image of 0.05, 0.1, 0.2, 0.5 noise attacks; (**e**–**h**) corresponding decrypted image of different noise attacks.

5.3.3. χ^2 Test

The chi-squared (χ^2) test reflects the quantized distribution of the image pixel distribution, which corresponds to the uniformity of the image histogram. A smaller chi-square value indicates a more uniform image histogram. The mathematical definition is as follows:

$$\chi^2 = \sum_{l=0}^{k} \frac{(O_l - f_e)^2}{f_e} \tag{16}$$

$$f_e = M \times N/256 \tag{17}$$

where k = 255, O_l denotes the actual observed frequency of occurrence of each gray level, f_e stands for the frequency of prediction. Table 13 illustrates the results of the chi-squared test and comparison with other similar algorithms. It can be reflected that the proposed algorithm has better image encryption performance.

Table 13. Comparison results of χ^2 test.

Algorithms	Images	χ^2 Test	
		Plain Image	Cipher Image
Proposed	Clock	282,062	244.914
	Couple	298,865	247.254
	Peppers	120,166	243.246
	Male	709,341	194.333
	Lena	158,349	185.645
	Boat	383,970	255.762
	Bridge	1,185,618	210.523
	Tank	2,025,900	249.018
	Baboon	187,357	254.293
[21]	Peppers	-	264.77
[30]	Peppers	-	287.22
[31]	Peppers	-	271.90
[24]	Peppers	-	253.08
[28]	Peppers	-	247.56
[15]	Lena	-	218
[32]	Lena	-	229
[37]	Lena	-	230
[38]	Lena	-	230
[39]	Lena	-	285

5.3.4. Information Entropy Analysis

The information entropy of an image is generally used to measure the amount of information contained in an image. The maximum value of information entropy is 8. The mathematical definition of information entropy is as follows:

$$Q = -\sum_{i=0}^{L} \rho(a_i) \log_2 \rho(a_i) \tag{18}$$

where $\rho(a_i)$ denotes the probability of occurrence of pixel a_i and L = 255. Table 14 lists the information entropies of different images and comparison results with other similar algorithms. It can be seen from Table 14 that the values of information entropies are extremely approximate 8, and the proposed algorithm has higher information entropies than that of other similar algorithms. Therefore, the proposed algorithm possesses enough security to defend against entropy attacks.

Table 14. Comparison results of information entropy.

Algorithms	Images	Information Entropy	
		Plain Image	Cipher Image
Proposed	Clock	6.7057	7.9973
	Couple	7.2010	7.9993
	Peppers	7.5937	7.9994
	Male	7.5237	7.9999
	Lena	7.4451	7.9995
	Boat	7.1914	7.9993
	Bridge	5.7056	7.9994
	Tank	5.4957	7.9993
	Baboon	7.3583	7.9993
[15]	Lena	-	7.9971
[25]	Lena	-	7.9971
[40]	Lena	-	7.9994
[41]	Lena	-	7.9993
[28]	Lena	-	7.9993
[24]	Peppers	-	7.9993
[28]	Peppers	-	7.9993
[15]	Peppers	-	7.9993
[19]	Peppers	-	7.9981
[25]	Peppers	-	7.9971

5.3.5. MSE and PSNR Analysis

The Mean Square Error (MSE) and peak signal-to-noise ratio are common metrics to evaluate the effect of image encryption. MSE and PSNR are defined as follows:

$$MSE = \frac{1}{S}\sum_{i=1}^{M}\sum_{j=1}^{N}(\rho(i,j) - \varsigma(i,j))^2 \tag{19}$$

$$PSNR = 10\log_{10}(\frac{L^2}{MSE}) \tag{20}$$

where M stands for the height of the image, and N stands for the width of the image, S = M × N. L = 255 represents the maximum gray level of the image. $\rho(i,j)$ denotes the original image and $\varsigma(i,j)$ denotes the cipher image or the encrypted image. For cipher images, a higher MSE value indicates a better image encryption effect, and a lower value of PSNR is better. For decrypted images, the value of PSNR represents the similarity between the decrypted image and the original image. MSE of the cipher image and PSNR of the cipher image and decrypted image are shown in Table 15.

Table 15. MSE and PSNR result.

Algorithms	Image	MSE	PSNR	
			Cipher	Decrypted
proposed	Clock	12,133.9	7.2791	∞
	Couple	7090.1	9.6085	∞
	Peppers	9428.4	8.3794	∞
	Male	10,288.3	7.9977	∞
	Lena	7754.2	9.2296	∞
	Boat	7628.4	9.3065	∞
	Bridge	8648.3	8.7615	∞
	Tank	6213.1	10.1977	∞
	Baboon	7285.1	9.5065	∞
[24]	Peppers	9267.0	8.4614	-
[24]	Male	10,294.6	8.0047	-
[24]	Couple	7089.2	9.6248	
[22]	Peppers	9311.7	8.4405	-
[22]	Lena	7752.6	9.2363	-

As is shown in Table 15, the proposed algorithm has higher MSE value and lower PSNR value than the other similar methods. The infinite PSNR value of the decrypted image indicates that the decrypted image is completely equivalent to the original image, which proves that the proposed algorithm is a lossless image algorithm.

5.3.6. Randomness Test

The randomness of the key in the image encryption process is directly related to image security, which is an important indicator of the security of the cryptosystem. We use the NIST SP800-22 test tool for randomness testing, which contains 16 test items. The p value of each item should be between 0.01 and 1 in the case of the pass.

In this paper, we test the randomness of the encryption key (KM, KE, KP, KX). The results of the NIST test are listed in Table 16. As is shown in Table 16, the full 16 NIST tests were passed, which proves that the encryption keys have good randomness.

Table 16. NIST test.

Test Items	Key: *KM*	Key: *KE*	Key: *KP*	Key: *KX*
Frequency test	0.991468	0.922325	0.863341	0.913568
Frequency test within a block	0.350485	0.350485	0.426843	0.269864
Cumulative sums test	0.739918	0.534146	0.634896	0.648576
Discrete fourier transform test	0.834308	0.622325	0.725563	0.268573

Table 16. *Cont.*

Test Items	Key: *KM*	Key: *KE*	Key: *KP*	Key: *KX*
Test for the longest run of ones in a block	0.911413	0.805189	0.415371	0.897631
Random excursions variant test (x = 1)	0.184287	0.911413	0.185632	0.213547
Runs test	0.637119	0.017912	0.223575	0.168573
Non-overlapping template matching test (m = 9)	0.995253	0.739918	0.279563	0.245982
Overlapping template matching test (m = 9)	0.929703	0.916905	0.958254	0.987024
Maurer's "Universal statistical" test	0.637683	0.720774	0.547635	0.635794
Approximate entropy test	0.231344	0.658603	0.446578	0.324359
Random excursions test (x = 1)	0.919708	0.773446	0.354783	0.738972
Serial test	0.833230	0.630451	0.885763	0.824629
Linear complexity test	0.888998	0.213309	0.514790	0.883671
Binary matrix rank test	0.350485	0.956353	0.965873	0.896542

5.4. Differential Attack Analyses

The attacker obtains key clues by comparing the difference between the original cipher image and the cipher image that fine-tunes the pixels of the original plain image, which is called differential attack. The Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are commonly used metrics to evaluate cryptosystems against differential attacks. The ideal values of NPCR and UACI are 99.6094% and 33.4635%. The mathematical expression is as follows:

$$NPCR(\lambda_1, \lambda_2) = \sum_{uv} \frac{D(u,v)}{W \times H} \times 100\% \tag{21}$$

$$D(u,v) = \begin{cases} 0, \lambda_1(u,v) = \lambda_2(u,v) \\ 1, \lambda_1(u,v) \neq \lambda_2(u,v) \end{cases} \tag{22}$$

$$UACI(\lambda_1, \lambda_2) = \frac{1}{W \times H} \left[\sum_{uv} \frac{|\lambda_1(u,v) - \lambda_2(u,v)|}{L_{\max}} \right] \times 100\% \tag{23}$$

where W and H stand for the width and height of the image, respectively, and $L_{\max} = 255$ represents the maximum gray level of the image. Furthermore, λ_1 denotes the original cipher image, and λ_2 represents the cipher image corresponding to changing the pixel of the plain image. From Table 17, it can be concluded that the values of NPCR and UACI are all extremely close to the ideal values, and our algorithm possesses better performance compared with other similar algorithms.

Table 17. NPCR and UACI.

Algorithms	Image	NPCR (%)	UACI (%)
proposed	Clock	99.6124	33.4044
	Couple	99.6326	33.4924
	Peppers	99.6101	33.4626
	Male	99.6132	33.4302
	Lena	99.6216	33.4556
	Boat	99.6159	33.4404
	Bridge	99.6197	33.4570
	Tank	99.5804	33.4169
	Baboon	99.6143	33.4834
[15]	Lena	99.6246	33.4115
[32]	Lena	99.6063	33.4477
[36]	Lena	99.636	33.465
[42]	Lena	99.6096	33.4645
[41]	Lena	99.65	33.61
[24]	Peppers	99.6101	33.4689
[30]	Peppers	99.6086	33.4398
[43]	Peppers	99.6068	33.4332
[44]	Peppers	99.6083	29.6188
[15]	Peppers	99.5922	33.3953

5.5. Encryption Quality Analyses

5.5.1. Encryption Quality (EQ)

Encryption Quality (EQ) is often used to describe the difference between the cipher image and the original image, which can be defined as follows:

$$EQ = \sum_{i=0}^{L} \left| h_{pi} - h_{ci} \right| / (L+1) \tag{24}$$

where L = 256 denotes 256 grayscale levels, and h_{p_i} and h_{c_i} denote the sequence of occurrence of gray value i of plain image and cipher image. The larger the difference degree, the better the image encryption quality. Table 18 lists the EQ value of different images. Table 19 demonstrates the result comparing EQ values. As shown in Table 19, the proposed algorithm obtains higher a EQ value than other similar algorithms. Therefore, our algorithm performs better image encryption quality.

Table 18. EQ value of different images.

Image	Clock	Couple	Peppers	Male	Lena	Boat	Bridge	Tank	Baboon	Airplane
EQ value	243.180	868.469	666.336	2325.789	664.695	866.664	1560.523	1483.602	775.031	1083.617

Table 19. The comparison of EQ value.

Algorithms	Clock	Peppers	Airplane
Proposed	243.180	666.336	1083
[24]	242	608	1083
[45]	244	156	303
[46]	243	156	302
[47]	243	158	300

5.5.2. Maximum Deviation (MD)

The maximum deviation is used to reflect maximum variance between the plain image and cipher image; similar to EQ, a higher value indicates a better image encryption effect of the image. MD is defined as follows:

$$MD = \frac{\left| h_{p_0} - h_{c_0} \right| + \left| h_{p_{255}} - h_{c_{255}} \right|}{2} + \sum_{i=1}^{254} \left| h_{p_i} - h_{c_i} \right| \tag{25}$$

where h_{p_i} denotes the sequence of occurrence of gray value i of the plain image, h_{c_i} represents the sequence of occurrence of gray value i of cipher image. The MD values of different images are shown in Table 20, and the comparison results of MD values are illustrated in Table 21. The experiment results demonstrate that the proposed encryption algorithm has a better image encryption effect than other algorithms.

Table 20. MD values of different images.

Image	Clock	Couple	Peppers	Male	Lena	Boat	Bridge	Tank	Baboon	Girl	Cameraman
MD value	62,006.5	221,745.5	144,003.5	580,419	169,140.5	220,897.5	398,970.5	378,793.5	197,379	59,322.5	64,374

Table 21. Comparison results of MD values.

Algorithms	Cameraman	Baboon	Girl
Proposed	64,374	207,379	59,322.5
[24]	255,613	201,300	58,741
[46]	63,920	46,540	49,588
[48]	49,129	17,652	48,639
[49]	38,912	55,129	39,187

5.6. Robustness Analyses

A robust image encryption algorithm is still able to ensure that the decrypted image is identifiable in the environment of cropping and noise attacks, which proves that the algorithm has good robustness.

5.6.1. Cropping Attack Analysis

When the image is under cropping attack, the decryption image quality will be affected. As a result, it is necessary to analyze the impact of cropping attack on decrypted image quality. In the experiment, we choose to mask part of the pixels of the cipher image and set the pixel value at the mask to 0, then we decrypted the masked cipher image. The cipher images with different cropping ratios (1/16, 1/4, 1/2, 3/4) and the corresponding decrypted images are shown in Figure 16. As is shown in Figure 16, when the cipher image is cropped by 1/16, 1/4 and 1/2, most of the effective information of the image can still be decrypted; even if the ciphertext image is cropped by 3/4, the salient contour features in the corresponding decrypted ciphertext image can still be identified. As a consequence, the proposed algorithm can effectively resist the cropping attack.

5.6.2. Noise Attack Analysis

In the process of image transmission, it is inevitable to be affected by noise. An excellent image encryption algorithm should ensure that the decrypted image is still identifiable in the case of noise attacks and avoid the image sender having to re-send the image to improve the efficiency of image transmission. Figure 17 illustrates the cipher images and corresponding decrypted images after adding different proportions (5%, 10%, 20%, 50%) of salt and pepper noise. As is shown in Figure 17, when 5%, 10% or 20% salt and pepper noise is added to the ciphertext image, the quality of the decrypted image is degraded, but the effective information in the image is only slightly affected. Even if the proportion reaches 50%, the overall contour information of the decrypted image is still recognizable. Therefore, our algorithm can efficiently defend against noise attack.

5.6.3. Classical Attack Analysis

There are the following four types of attack patterns in cryptology.

Known-plaintext attack: The attacker steals part of the plaintext information and corresponding ciphertext information and obtains the key by analyzing the relationship between the known plaintext and ciphertext.

Ciphertext-only attack: The attacker only obtains the decrypted information (ciphertext information) to attack.

Chosen-plaintext attack: The attacker not only knows the encryption algorithm but also can construct specific ciphertext information. As the system can construct arbitrary ciphertext information, the attacker can obtain the key used by the cryptosystem more quickly and efficiently.

Chosen-ciphertext attack: The attacker not only knows the decryption algorithm but also can construct specific plaintext information.

Among these four attack modes, chosen plaintext attack is the most effective pattern. If a cryptosystem can defend against one chosen plaintext attack, it can naturally resist the other three attack modes.

The proposed algorithm uses SHA-256 to generate 32 decimal hash values related to the plaintext, and these 32 hash values are used to determine the initial value of the hyper-chaotic system, as well as the four convolution kernels of the convolutional neural network. At the same time, the convolutional neural network is used to generate convolution sequences for controlling the process of DNA strand permutation. Therefore, different

images will generate different hyper-chaotic sequences and convolution sequences, which will lead to different encryption and decryption results. In other words, our algorithm is a complete one-time pad encryption and decryption system, which can efficiently defend against the chosen-plaintext attack and the other three attacks above.

5.7. Encryption Efficiency Analyses

5.7.1. Execution Time Analysis

Algorithm execution time is an essential index to measure the performance and efficiency of the cryptosystem. An excellent image encryption algorithm can ensure security while requiring as little execution time as possible. The comparison result of execution time analysis is shown in Table 22. Above all, Table 22 lists the execution times of encryption and decryption for three different sizes of images; our algorithm shows shorter encryption and decryption times than the algorithm in [22]. In addition, we also compare the encryption time of the size of the 512 × 512 image with the other similar method in [50–53], which also demonstrate the efficiency of the proposed algorithm compared with other similar algorithms. At the same time, the execution time of every image encryption step is listed in Table 23. As is shown in Table 23, hyper-chaotic and convolution sequences generation steps account for 60% of the total execution time. However, the DNA strand generation step and DNA strand permutation and diffusion steps only take up 5% and 6% of the total time, which demonstrates the efficiency of the bit-plane decomposition method to generate DNA strands and the DNA scrambling and diffusion operations.

Table 22. Comparison results of execution time analysis.

Algorithm	Image Size	Encryption Time	Decryption Time	Platform & Hardware
Proposed	256 × 256	0.3280	0.3944	MATLAB R2022a, CPU i7&2.80GHz, 8G RAM
	512 × 512	1.3650	1.5832	
	1024 × 1024	5.3446	6.6094	
[22]	256 × 256	0.380	0.463	MATLAB R2020a, CPU i5&2.80GHz, 8G RAM
	512 × 512	1.431	1.633	
	1024 × 1024	5.719	6.759	
[50]	512 × 512	1.6455	-	MATLAB R2013a, CPU i3&2.4 GHz, 4G RAM
[51]		23.142	-	MATLAB R2018b, CPU i7&3.4 GHz, 8G RAM
[52]		6.8816	-	MATLAB R2017a, CPU i5&2.3 GHz, 8G RAM
[53]		6.0752	-	MATLAB R2017a, CPU i5&2.8 GHz, 8G RAM

Table 23. Execution time of every encryption step.

Steps	Time (/s)	Percentage
Hyper-chaotic sequence generation	0.8012	59%
Convolution sequence generation	0.0178	1%
Bit-plane decomposition	0.0332	2%
DNA strand generation	0.0677	5%
DNA strand encoding	0.0518	4%
DNA strand permutation	0.0603	4%
DNA strand diffusion	0.0314	2%
DNA strand decoding Bit-plane combination	0.2841	21%
Other	0.0175	1%
Total	1.3650	100%

5.7.2. Time Complexity Analysis

Time complexity analysis is obtained by summing the time complexity of each major encryption process. m × n stands for the total number of pixels. The major encryption processes for calculating the time complexity include hyper-chaotic and convolution sequence generation, DNA strand generation, DNA strand permutation, DNA strand diffusion, DNA strand encoding and decoding and bit-plane combined. The time complexity of hyper-chaotic and convolution sequence generation is O (10 × m × n). For the DNA strand generation, the bit-plane decomposition is essentially a subprocess, and the time complexity is O (4 × m × n). The time complexity of DNA strand permutation is O (4 × m × n). The time complexity of DNA strand diffusion is O (4 × m × n). The time complexity of DNA

strand diffusion is O(4 × m × n). For the DNA strand encoding and decoding, the time complexity is O (4 × m × n). In addition, the bit-plane combination first converts the DNA strands into binary sequences, then splits the sequences and finally merges them. The time complexity of bit-plane combination is O (m × n). As a consequence, the overall time complexity is O (m × n), which is less than O (37 × m × n) and O (78 × m × n) in [24].

6. Conclusions

In this paper, a DNA encoding image encryption algorithm based on convolution is proposed. We designed a novel DNA strand generation method based on bit-plane decomposition and DNA encoding. According to the operation rules of DNA operation, we construct seven DNA operation index tables. By retrieving the index table, the DNA operation process is simplified and accelerated, which ensures the diffusion efficiency of the DNA strand and the effect of image encryption. Simulation results and experimental analysis show that the proposed encryption scheme exhibits superior security, effectiveness and robustness.

Although the five-dimensional Hamiltonian conservative chaotic system has good dynamic behavior, its chaotic sequence generation time accounts for 59% of the total execution time, which still has huge room for improvement. In future work, we will explore more efficient chaotic sequence generation schemes to further improve the encryption efficiency of the cryptosystem.

Author Contributions: The authors confirm their contribution to the paper as follows: Conceptualization, L.H. and C.D.; methodology, C.D.; software, C.D.; validation, L.H., C.W. and Z.B.; formal analysis, L.H.; investigation, L.H.; resources, H.C.; data curation, L.H.; writing—original draft preparation, L.H.; writing—review and editing, C.D.; visualization, L.H.; supervision, H.C.; project administration, L.H.; funding acquisition, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Key Laboratory of Specialty Fiber Optics and Optical Access Networks, grant number SKLSFO2024-04.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to privacy.

Acknowledgments: The authors would like to thank the editor and anonymous reviewers for their valuable comments that helped to improve this paper.

Conflicts of Interest: Author Haitao Chen was employed by the company Hunan Zhentong Zhiyong Artificial Intelligence Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Lorenz, E.N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [CrossRef]
2. Tufillaro, N.B. Nonlinear and chaotic string vibrations. *Am. J. Phys.* **1989**, *57*, 408–414. [CrossRef]
3. Matthews, R. On the derivation of a "chaotic" encryption algorithm. *Cryptologia* **1989**, *13*, 29–42. [CrossRef]
4. Jain, A.; Rajpal, N. A robust image encryption algorithm resistant to attacks using DNA and chaotic logistic maps. *Multimed. Tools Appl.* **2015**, *75*, 5455–5472. [CrossRef]
5. Chai, X.-L.; Gan, Z.-H.; Lu, Y.; Zhang, M.-H.; Chen, Y.-R. A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system. *Chin. Phys. B* **2016**, *25*, 100503. [CrossRef]
6. Wu, X.; Wang, D.; Kurths, J.; Kan, H. A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system. *Inf. Sci.* **2016**, *349*, 137–153. [CrossRef]
7. Shakir, H.R. An image encryption method based on selective AES coding of wavelet transform and chaotic pixel shuffling. *Multimed. Tools Appl.* **2019**, *78*, 26073–26087. [CrossRef]

8. Gong, L.; Qiu, K.; Deng, C.; Zhou, N. An image compression and encryption algorithm based on chaotic system and compressive sensing. *Opt. Laser Technol.* **2019**, *115*, 257–267. [CrossRef]
9. Wang, H.; Xiao, D.; Li, M.; Xiang, Y.; Li, X. A visually secure image encryption scheme based on parallel compressive sensing. *Signal Process.* **2019**, *155*, 218–232. [CrossRef]
10. Xu, Q.; Sun, K.; Cao, C.; Zhu, C. A fast image encryption algorithm based on compressive sensing and hyperchaotic map. *Opt. Lasers Eng.* **2019**, *121*, 203–214. [CrossRef]
11. Luo, Y.; Tang, S.; Liu, J.; Cao, L.; Qiu, S. Image encryption scheme by combining the hyper-chaotic system with quantum coding. *Opt. Lasers Eng.* **2020**, *124*, 105836. [CrossRef]
12. Huang, H.; Yang, S.; Ye, R. Efficient symmetric image encryption by using a novel 2D chaotic system. *IET Image Process.* **2020**, *14*, 1157–1163. [CrossRef]
13. Hua, Z.; Zhang, K.; Li, Y.; Zhou, Y. Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing. *Signal Process.* **2021**, *183*, 107998. [CrossRef]
14. Hua, Z.; Li, J.; Chen, Y.; Yi, S. Design and application of an S-box using complete Latin square. *Nonlinear Dyn.* **2021**, *104*, 807–825. [CrossRef]
15. Zou, C.; Wang, X.; Zhou, C.; Xu, S.; Huang, C. A novel image encryption algorithm based on DNA strand exchange and diffusion. *Appl. Math. Comput.* **2022**, *430*, 127291. [CrossRef]
16. Liu, X.; Tong, X.; Zhang, M.; Wang, Z.; Fan, Y. Image compression and encryption algorithm based on uniform non-degeneracy chaotic system and fractal coding. *Nonlinear Dyn.* **2023**, *111*, 8771–8798. [CrossRef]
17. Dong, E.; Yuan, M.; Du, S.; Chen, Z. A new class of Hamiltonian conservative chaotic systems with multistability and design of pseudo-random number generator. *Appl. Math. Model.* **2019**, *73*, 40–71. [CrossRef]
18. Dong, Q.; Zhou, S.; Zhang, Q.; Kasabov, N.K. A class of 5D Hamiltonian conservative hyperchaotic systems with symmetry and multistability. *Nonlinear Dyn.* **2022**, *110*, 2889–2912. [CrossRef]
19. Man, Z.; Li, J.; Di, X.; Sheng, Y.; Liu, Z. Double image encryption algorithm based on neural network and chaos. *Chaos Solitons Fractals* **2021**, *152*, 111318. [CrossRef]
20. Patel, S.; Bharath, K.P.; Rajesh Kumar, M. Symmetric keys image encryption and decryption using 3D chaotic maps with DNA encoding technique. *Multimed. Tools Appl.* **2020**, *79*, 31739–31757. [CrossRef]
21. Wang, X.; Zhao, M. An image encryption algorithm based on hyperchaotic system and DNA coding. *Opt. Laser Technol.* **2021**, *143*, 107316. [CrossRef]
22. Jasra, B.; Hassan Moon, A. Color image encryption and authentication using dynamic DNA encoding and hyper chaotic system. *Expert Syst. Appl.* **2022**, *206*, 117861. [CrossRef]
23. Wang, Q.Y.; Zhang, X.Q.; Zhao, X.H. Image encryption algorithm based on improved Zigzag transformation and quaternary DNA coding. *J. Inf. Secur. Appl.* **2022**, *70*, 18. [CrossRef]
24. Elizabeth, B.L.; Gayathri, J.; Subashini, S.; Prakash, A.J. HIDE: Hyperchaotic image encryption using DNA computing. *J. Real-Time Image Process.* **2022**, *19*, 429–443. [CrossRef]
25. Liu, Y.; Zhang, J. A Multidimensional Chaotic Image Encryption Algorithm based on DNA Coding. *Multimed. Tools Appl.* **2020**, *79*, 21579–21601. [CrossRef]
26. Yu, W.; Liu, Y.; Gong, L.; Tian, M.; Tu, L. Double-image encryption based on spatiotemporal chaos and DNA operations. *Multimed. Tools Appl.* **2019**, *78*, 20037–20064. [CrossRef]
27. Lin, R.; Li, S.; Tan, Z. An Image Encryption Scheme Based on Lorenz Hyperchaotic System and RSA Algorithm. *Secur. Commun. Netw.* **2021**, *2021*, 5586959. [CrossRef]
28. Zhang, Y.Q.; Wang, X.Y. A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. *Inf. Sci.* **2014**, *273*, 329–351. [CrossRef]
29. Ye, G.; Wu, H.; Jiao, K.; Mei, D. Asymmetric image encryption scheme based on the Quantum logistic map and cyclic modulo diffusion. *Math. Biosci. Eng.* **2021**, *18*, 5427–5448. [CrossRef]
30. Patro, K.A.K.; Soni, A.; Netam, P.K.; Acharya, B. Multiple grayscale image encryption using cross-coupled chaotic maps. *J. Inf. Secur. Appl.* **2020**, *52*, 102470. [CrossRef]
31. Shengtao, G.; Tao, W.; Shida, W.; Xuncai, Z.; Ying, N. A Novel Image Encryption Algorithm Based on Chaotic Sequences and Cross-Diffusion of Bits. *IEEE Photonics J.* **2021**, *13*, 3900215. [CrossRef]
32. Roy, M.; Chakraborty, S.; Mali, K.; Roy, D.; Chatterjee, S. A robust image encryption framework based on DNA computing and chaotic environment. *Microsyst. Technol.* **2021**, *27*, 3617–3627. [CrossRef]
33. Zhang, Y.-Q.; Huang, H.-F.; Wang, X.-Y.; Huang, X.-H. A secure image encryption scheme based on genetic mutation and MLNCML chaotic system. *Multimed. Tools Appl.* **2021**, *80*, 19291–19305. [CrossRef]
34. Hu, G.; Li, B. A uniform chaotic system with extended parameter range for image encryption. *Nonlinear Dyn.* **2021**, *103*, 2819–2840. [CrossRef]

35. Wang, T.; Wang, M.-h. Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding. *Opt. Laser Technol.* **2020**, *132*, 106355. [CrossRef]
36. Alghafis, A.; Firdousi, F.; Khan, M.; Batool, S.I.; Amin, M. An efficient image encryption scheme based on chaotic and Deoxyribonucleic acid sequencing. *Math. Comput. Simul.* **2020**, *177*, 441–466. [CrossRef]
37. Wang, X.; Feng, L.; Zhao, H. Fast image encryption algorithm based on parallel computing system. *Inf. Sci.* **2019**, *486*, 340–358. [CrossRef]
38. Wang, X.; Liu, C.; Xu, D.; Liu, C. Image encryption scheme using chaos and simulated annealing algorithm. *Nonlinear Dyn.* **2016**, *84*, 1417–1429. [CrossRef]
39. Wang, X.; Liu, P. Image encryption based on roulette cascaded chaotic system and alienated image library. *Vis. Comput.* **2021**, *38*, 763–779. [CrossRef]
40. Zheng, J.; Hu, H. A symmetric image encryption scheme based on hybrid analog-digital chaotic system and parameter selection mechanism. *Multimed. Tools Appl.* **2021**, *80*, 20883–20905. [CrossRef]
41. Zhang, Q.; Han, J.; Ye, Y. Multi-image encryption algorithm based on image hash, bit-plane decomposition and dynamic DNA coding. *IET Image Process.* **2021**, *15*, 885–896. [CrossRef]
42. Shakiba, A. A novel randomized bit-level two-dimensional hyperchaotic image encryption algorithm. *Multimed. Tools Appl.* **2020**, *79*, 32575–32605. [CrossRef]
43. Zhang, Y. The unified image encryption algorithm based on chaos and cubic S-Box. *Inf. Sci.* **2018**, *450*, 361–377. [CrossRef]
44. Gayathri, J.; Subashini, S. A spatiotemporal chaotic image encryption scheme based on self adaptive model and dynamic keystream fetching technique. *Multimed. Tools Appl.* **2018**, *77*, 24751–24787. [CrossRef]
45. Wang, X.; Zhang, M. An image encryption algorithm based on new chaos and diffusion values of a truth table. *Inf. Sci.* **2021**, *579*, 128–149. [CrossRef]
46. Farah, M.; Guesmi, R.; Kachouri, A.; Samet, M. A new design of cryptosystem based on S-box and chaotic permutation. *Multimed. Tools Appl.* **2020**, *79*, 19129–19150. [CrossRef]
47. Ahmed, F.; Anees, A.; Abbas, V.U.; Siyal, M.Y. A Noisy Channel Tolerant Image Encryption Scheme. *Wirel. Pers. Commun.* **2014**, *77*, 2771–2791. [CrossRef]
48. Anees, A.; Siddiqui, A.M.; Ahmed, F. Chaotic substitution for highly autocorrelated data in encryption algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 3106–3118. [CrossRef]
49. Yousif, S.F.; Abboud, A.J.; Alhumaima, R.S. A new image encryption based on bit replacing, chaos and DNA coding techniques. *Multimed. Tools Appl.* **2022**, *81*, 27453–27493. [CrossRef]
50. Huang, Z.-W.; Zhou, N.-R. Image encryption scheme based on discrete cosine Stockwell transform and DNA-level modulus diffusion. *Opt. Laser Technol.* **2022**, *149*, 107879. [CrossRef]
51. Yan, X.; Wang, X.; Xian, Y. Chaotic image encryption algorithm based on arithmetic sequence scrambling model and DNA encoding operation. *Multimed. Tools Appl.* **2021**, *80*, 10949–10983. [CrossRef]
52. Wang, X.; Zhang, M. A new image encryption algorithm based on ladder transformation and DNA coding. *Multimed. Tools Appl.* **2021**, *80*, 13339–13365. [CrossRef]
53. ElKamchouchi, D.H.; Mohamed, H.G.; Moussa, K.H. A Bijective Image Encryption System Based on Hybrid Chaotic Map Diffusion and DNA Confusion. *Entropy* **2020**, *22*, 180. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.