entropy

MDPI

Article

Hyper-Chaotic Color Image Encryption Based on Transformed Zigzag Diffusion and RNA Operation

Duzhong Zhang, Lexing Chen and Taiyong Li *

School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 611130, China; zhangduzhong@swufe.edu.cn (D.Z.); clx220081203001@smail.swufe.edu.cn (L.C.)
* Correspondence: litaiyong@gmail.com

Abstract: With increasing utilization of digital multimedia and the Internet, protection on this digital information from cracks has become a hot topic in the communication field. As a path for protecting digital visual information, image encryption plays a crucial role in modern society. In this paper, a novel six-dimensional (6D) hyper-chaotic encryption scheme with three-dimensional (3D) transformed Zigzag diffusion and RNA operation (HCZRNA) is proposed for color images. For this HCZRNA scheme, four phases are included. First, three pseudo-random matrices are generated from the 6D hyper-chaotic system. Second, plaintext color image would be permuted by using the first pseudo-random matrix to convert to an initial cipher image. Third, the initial cipher image is placed on cube for 3D transformed Zigzag diffusion using the second pseudo-random matrix. Finally, the diffused image is converted to RNA codons array and updated through RNA codons tables, which are generated by codons and the third pseudo-random matrix. After four phases, a cipher image is obtained, and the experimental results show that HCZRNA has high resistance against well-known attacks and it is superior to other schemes.

Keywords: hyper-chaotic; ribonucleic acid; color image encryption; transformed Zigzag

Citation: Zhang, D.; Chen, L.; Li, T. Hyper-Chaotic Color Image Encryption Based on Transformed Zigzag Diffusion and RNA Operation. *Entropy* **2021**, *23*, 361. https://doi.org/10.3390/e23030361

Academic Editors: Amelia Carolina Sparavigna and Salim Lahmiri

Received: 27 January 2021
Accepted: 15 March 2021
Published: 17 March 2021

1. Introduction

Nowadays, rapid developments of Internet and digital technologies have led to tremendous digital multimedia contents transmitting over Internet networks. Thus, protection on the contents of digital data has attracted serious concern from medical, military, and many other areas. Various image encryption methods have emerged by using cryptographic techniques [1–4]. Although there exists a view that AES is not suitable for image encryption, Zhang recently refuted it by using AES of cipher block chaining mode to encrypt images [5].

The chaos-based encryption method has become one of the most ideal methods, since it has a lot of appropriate characteristics, e.g. high sensitivity on initial conditions, mixing property, ergodicity, complex behavior, etc. [6–8]. As a result, a lot of researchers have presented plenty of image encryption schemes with a chaotic system [9–13]. In [14], Askar et al. proposed a chaotic economic map based image encryption method, whose simulation results indicated that the proposed algorithm could successfully encrypt and decrypt the images, and it had a good performance on security tests, except noise attacks analysis. By using a single round based hyper-chaotic system, Shaikh et al. presented a color image encryption method with bi-directional pixel diffusion [15]. Additionally, Li et al. presented a "transforming-scrambling-diffusion" model based color image encryption method with a four-dimensional (4D) hyper-chaotic system, which could convert pixel values to gray format before scrambling [16]. There is no doubt that some of the encryption methods in these chaos-based schemes still have weaknesses to some extent. However, different chaotic systems are neither superior nor inferior each other. A high-dimensional chaotic system has complex chaotic behaviors with high time cost, while a low-dimensional

chaotic system is opposite [17–19]. Hence, in this paper, a 6D hyper-chaotic system is employed as a pseudo-random numbers sequence generator for more complexity.

Zigzag is a common scrambling operation in image encryption [20,21]. In [22], Li et al. presented a 3D logistic map based color image encryption method with Zigzag scramble; the experiments showed that this method had brute-force attack and statistical attack resistance, but differential attacks analysis was missing. While, Wang et al. proposed a color encryption method with a Zigzag transformation, which could change the start pixel from upper left corner to the other three corners in an image [20]. Next year, Wang et al. [23] presented another image encryption method, which introduced an extended Zigzag confusion for a non-square image. Additionally, in [24], Zhao et al. proposed a novel color image encryption by combining Zigzag map and Hénon map together for permutation. However, these image encryption schemes implement Zigzag scramble on 2D images, which leads to some adjacent values in special positions of the image not being able to be scrambled, and different channels of a color image could not be scrambled, either. On the other hand, some image encryptions transformed 2D image to 3D cube [25], which gives out a new encryption inspiration on permutation, but most of them were focused on rotation, but not Zigzag. Therefore, Zigzag is utilized in diffusion on a 3D cube instead of scramble on 2D image to eliminate these drawbacks in this paper.

Deoxyribonucleic acid (DNA), a biological concept, has recently become a popular trend in the image encryption field [26,27]. By using DNA-based techniques, cipher images could obtain competitive entropy, correlation coefficients etc. [4,28–31]. In [29], Chai et al. presented a new diffusion mechanism that is based on the random numbers that are generated by plaintext image, and incorporated DNA encryption with four-wing hyper-chaotic system. Reference [32] proposed an image encryption method using a spatial map based DNA sequence matrix. In general, the DNA-based encryption mechanism includes two steps: use DNA operation rules to convert pixels of plaintext image to DNA codon matrix and change chaotic sequence to DNA keys to generate cipher image with DNA codon matrix.

While unlike the two strands structure of DNA sequences, Ribonucleic acid (RNA) is a single strand structure. RNA could form double helixes with complementary base pairing. By using this feature, some new image encryption methods have been proposed. In [33], Mahmud et al. presented an image encryption method by combining RNA with Genetic Algorithm (GA) through using a logistic map. In [34], Abbasi et al. employed Chen's chaotic system to encrypt an image with imperialist competition algorithm and RNA operations. Yadollahi et al. utilized the concepts of DNA and RNA to construct a two-phase image encryption method [35]. While an image encryption method is presented by Wang et al. through using an one-dimensional (1D) chaotic system combined from Logistic and Sine map, extended Zigzag confusion, and RNA operation [23]. However, all of these four schemes focus on gray image encryption. Although there is a color image experiment in [23], it is realized by running the scheme three times in three channels.

Being motivated by above discussions, a novel color image encryption method, called HCZRNA, is proposed in this paper. At the beginning, a 6D hyper-chaotic system is employed to generate three pseudo-random matrices. Subsequently, one of the pseudo-random matrices is used to permute plaintext color image. Additionally, 3D transformed Zigzag diffusion is implemented on initial cipher image with the second pseudo-random matrix. After diffusion, an RNA operation is used to convert the diffused image to RNA codons array, and update this array through RNA codons tables that are generated by the third pseudo-random matrix. Finally, a cipher image is obtained.

The main contributions of this work is listed as follows:

- A novel 6D hyper-chaotic system is employed in this paper to produce chaotic matrix for permutation, diffusion, and RNA operation.
- A new 3D transformed Zigzag diffusion scheme is proposed to encrypt color images.
- RNA operation is modified specifically for color images.

- Extensive experiments and analyses demonstrate that the proposed HCZRNA could resist various types of attacks.

The rest of this paper is structured as follows: Section 2 introduces the used 6D hyper-chaotic system, 3D Zigzag and RNA. Section 3 presents the HCZRNA scheme and explains how initial values and pseudo-random matrix are generated in detail. Section 4 reports and analyzes the experimental results. Finally, Section 5 concludes this paper.

2. Preliminaries

2.1. The 6D Hyper-Chaotic System

There are a lot of classical chaotic systems, e.g. Sine map, Logistic map, Tent map, etc., which have simple mathematical forms and can be implemented easily. However, they suffer from small key spaces, predictable orbits, limited ranges, etc. Existing research has shown that higher dimensional chaotic systems are much securer for image encryption [36]. Therefore, a novel 6D hyper-chaotic system is employed in this paper for chaotic sequences generation, which could be described as Equation (1) [37].

$$\begin{aligned} \dot{x}_1 &= g(\omega + \beta x_6^2)x_2 - ax_1 \\ \dot{x}_2 &= cx_1 + dx_2 - x_1x_3 + x_5 \\ \dot{x}_3 &= -bx_3 + x_1^2 \\ \dot{x}_4 &= ex_2 + fx_4 \\ \dot{x}_5 &= -rx_1 \\ \dot{x}_6 &= x_2 \end{aligned} \tag{1}$$

where $a, b, c, d, e, f, g, r, \omega$, and β are controlling parameters, and $x_i (i = 1, 2, \cdots, 6)$ are state variables.

The fourth-order Runge–Kutta method is used to solve this hyper-chaotic system with step size $h = 0.001$. We set the controlling parameters as $(a, b, c, d, e, f, g, r, \omega, \beta) = (0.3, 1.5, 8.5, -2, 1, -0.1, 0.9, 1, 1, 0.2)$ and initial state variables as $(x_1, x_2, x_3, x_4, x_5, x_6) = (0.1, 0.6, 0.2, 0.02, 1, 0.5)$; Figure 1 shows this 6D hyper-chaotic system's attractors. Its Lyapunov exponents are $\lambda_1 = 7.340$, $\lambda_2 = 0.087$, $\lambda_3 = 0.006$, $\lambda_4 = -0.368$, $\lambda_5 = -1.349$, $\lambda_6 = -67.426$. Since this chaotic system has three positive Lyapunov exponents, its prediction time should be longer than other chaotic systems and it is hard to crack. Besides, this hyper-chaotic system exhibits limit cycles, quasiperiodic, and bursting behavior. Accordingly, it could generate effective a pseudo random sequence. More detailed demonstration could be found in reference [37].

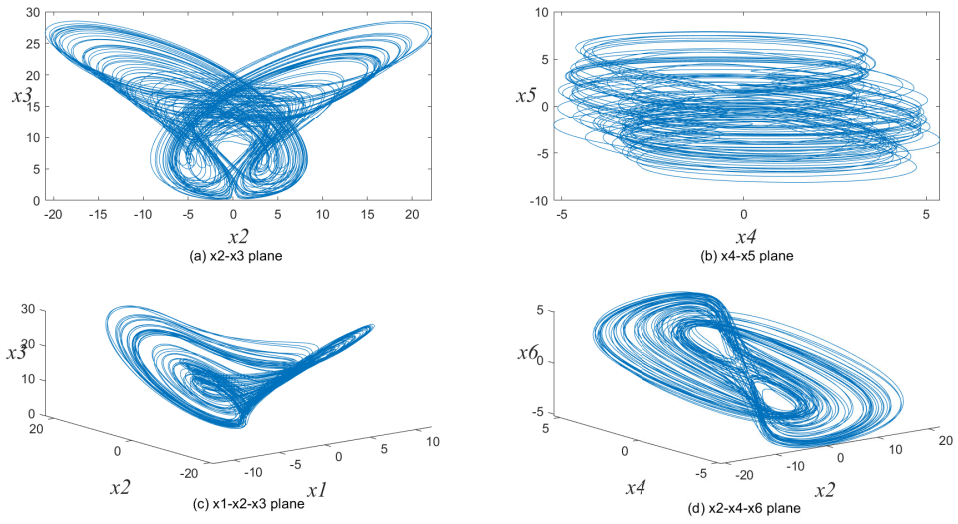


Figure 1. The attractors of six-dimensional (6D) hyper-chaotic system.

2.2. 3D Transformed Zigzag Diffusion

Zigzag is a scanning method that is used to scramble pixels in image encryption. By scanning and taking a pixel with left upper corner of image, then taking other pixels one-by-one through Zigzag path, the image could be converted to a matrix in a fixed way. Hence, the image's pixels could be scrambled.

Traditional Zigzag scrambling could only walk through numbers in $N \times N$ matrix with a fixed Zigzag path; an example of 4×4 matrix is shown in Figure 2. By this fixed path, different channels of color image could not be scrambled with each other. Due to its drawbacks, this paper proposes a novel 3D Zigzag transformation. Using this transformation, each channel of color image would be cut into two triangles through a diagonal line, and be placed on opposite surfaces of a cube, which is illustrated by an example of $4 \times 4 \times 3$ matrix, as shown in Figure 3. Subsequently, diffusion would start from origin vertex of the frontal side of the cube, and walk through every pixels on six surfaces at the front and back side synchronously with spiral Zigzag path, as in Figure 4. For all triangles that are placed on the cube, the order of diffusion is shown in Figure 5. In this way, different channels of color image could be diffused together.

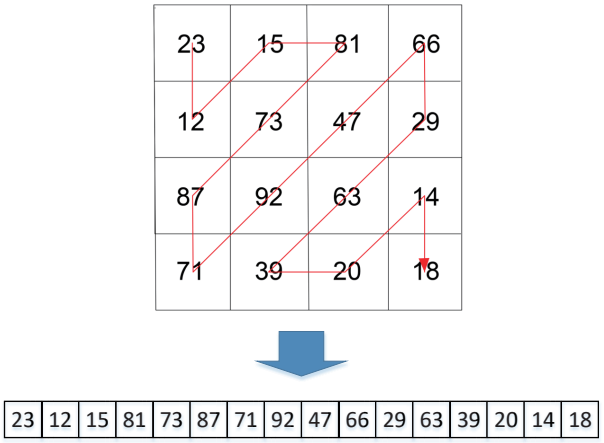


Figure 2. 4×4 traditional Zigzag scramble.

2.3. RNA Operation

RNA is one of the major macromolecules necessary for living organism. RNA has a single strand structure with four nitrogen bases: adnine (A), cytosine (C), guanine (G), and uracil (U). For these four units of RNA, a binary system could be employed for representation, which is shown in Table 1. According to the base pairing rules, four bases of RNA could be coded and constructed into three nucleotides that correspond to one amino acid called codon. Accordingly, there are 64 codons truth table of bases combinations, as shown in Table 2. Assuming that pixels in the image could transfer into six-bits format, a corresponding RNA codon could be found in Table 2.

Table 1. Binary representation of RNA.

RNA Bases	A	C	G	U
Binary	00	01	10	11

Table 2. RNA codon table.

#	Bin.	Codon	#	Bin.	Codon	#	Bin.	Codon	#	Bin.	Codon
0	000000	AAA	16	010000	CAA	32	100000	GAA	48	110000	UAA
1	000001	AAC	17	010001	CAC	33	100001	GAC	49	110001	UAC
2	000010	AAG	18	010010	CAG	34	100010	GAG	50	110010	UAG
3	000011	AAU	19	010011	CAU	35	100011	GAU	51	110011	UAU
4	000100	ACA	20	010100	CCA	36	100100	GCA	52	110100	UCA
5	000101	ACC	21	010101	CCC	37	100101	GCC	53	110101	UCC
6	000110	ACG	22	010110	CCG	38	100110	GCG	54	110110	UCG
7	000111	ACU	23	010111	CCU	39	100111	GCU	55	110111	UCU
8	001000	AGA	24	011000	CGA	40	101000	GGA	56	111000	UGA
9	001001	AGC	25	011001	CGC	41	101001	GGC	57	111001	UGC
10	001010	AGG	26	011010	CGG	42	101010	GGG	58	111010	UGG
11	001011	AGU	27	011011	CGU	43	101011	GGU	59	111011	UGU
12	001100	AUA	28	011100	CUA	44	101100	GUA	60	111100	UUA
13	001101	AUC	29	011101	CUC	45	101101	GUC	61	111101	UUC
14	001110	AUG	30	011110	CUG	46	101110	GUG	62	111110	UUG
15	001111	AUU	31	011111	CUU	47	101111	GUU	63	111111	UUU

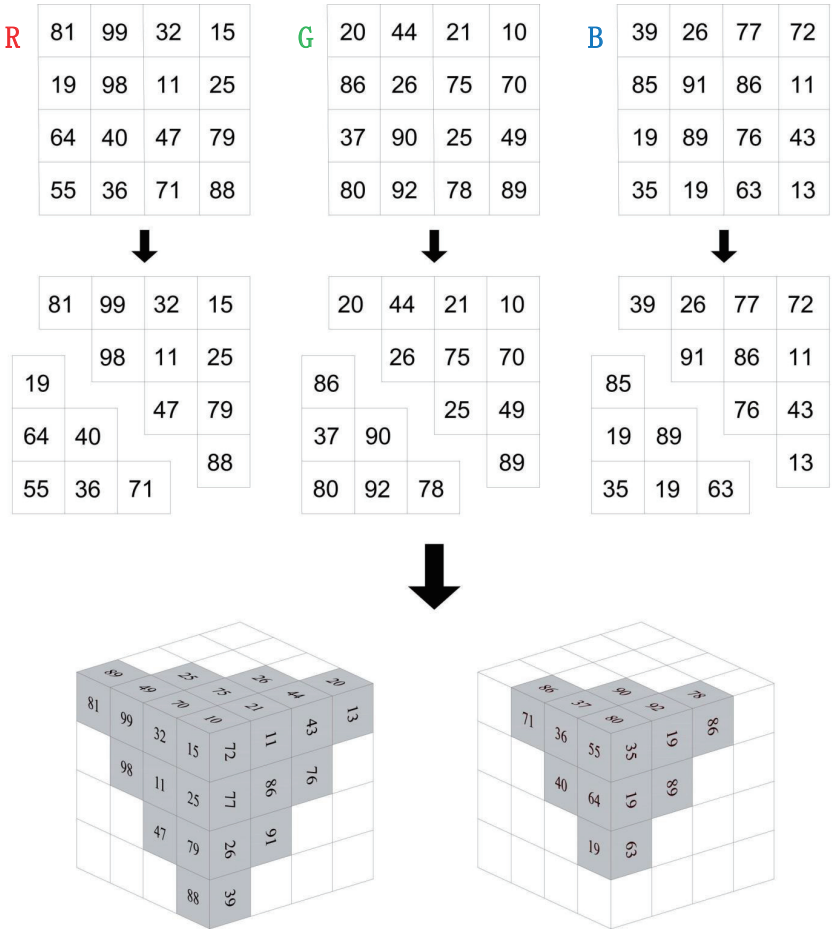
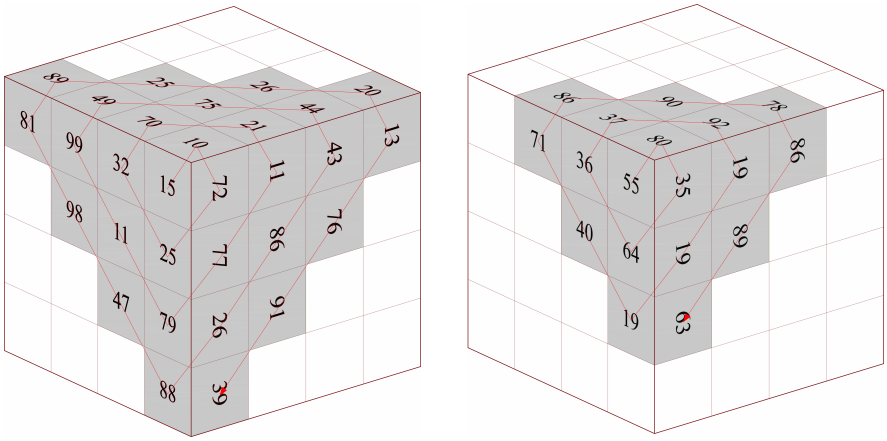


Figure 3. Color image to cube.The first row is three channels of a color image. The second row is the triangles generated from image. Additionally, the third row is the placement of triangles on a cube.



(a) The front side (b) The back side

Figure 4. Three-dimensional (3D) transformed Zigzag diffusion. (**a**) is the Zigzag diffusion process on the front side of cube. (**b**) is the Zigzag diffusion process on the back side of cube.

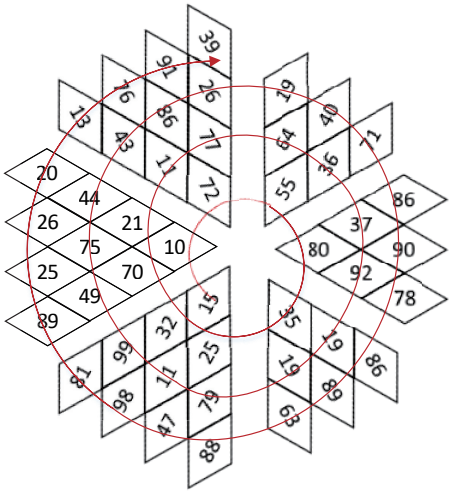


Figure 5. 3D transformed Zigzag path. For all triangles on the cube, 3D transformed Zigzag diffusion is implemented through this order.

3. Encryption and Decryption

In this paper, image encryption could be divided into three parts. Firstly, a 6D hyper-chaotic system is employed to generate chaotic matrices for encryption processes. Subsequently, three-dimensional (3D) transformed Zigzag diffusion is implemented on the permuted image. Finally, RNA concept is used for encoding and decoding.

3.1. Encryption Scheme

Suppose that plaintext image has N rows and N columns with RGB channels.

The flowchart of HCZRNA is described in Figure 6, and the specific operations are listed, as follows.

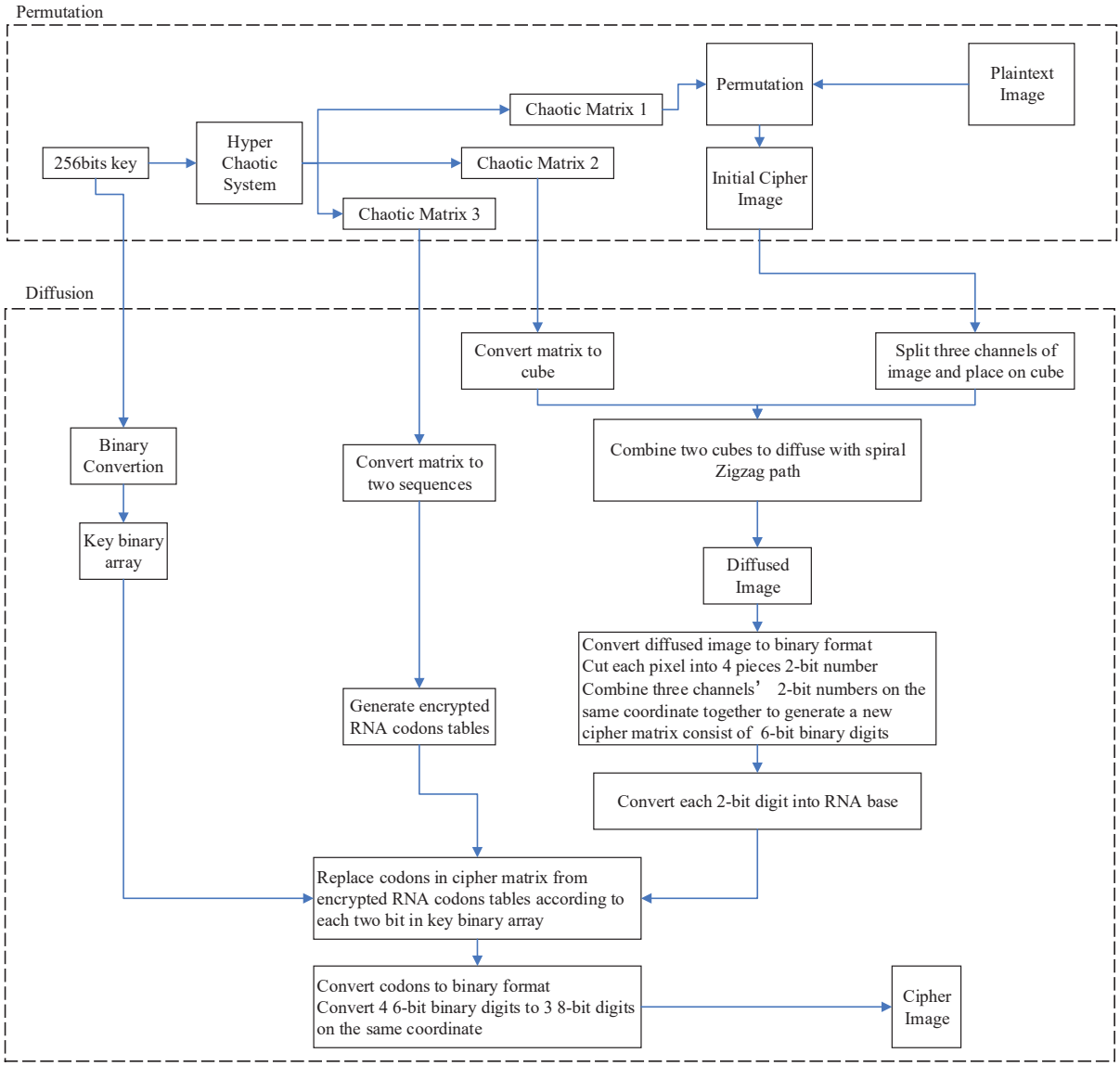


Figure 6. The process of encryption.

3.1.1. Initial Values Generation

The HCZRNA scheme uses a 256-bit key of different characters against attacks. The 256-bit long security key would be utilized in two parts, which are hyper-chaotic system initial values generation and RNA encryption.

At first, the initial values of hyper-chaotic system should be generated by a security key. Details of initial values generation is performed in three steps:

- Step 1: divide the secret key K into 32 blocks, which could be expressed as $K = \{k_1, k_2, \ldots, k_{32}\}$, each k is a 8-bits number.

- Step 2: K array that is generated in step 1 is calculated into four intermediate parameters d_1, d_2, d_3, d_4 by Equation (2) with four user-defined constants c_1, c_2, c_3 and c_4.

$$\begin{cases} d_1 = c_1 + \dfrac{k_1 \oplus k_2 \oplus \cdots \oplus k_8}{256} \\ d_2 = c_2 + \dfrac{k_9 \oplus k_{10} \oplus \cdots \oplus k_{16}}{256} \\ d_3 = c_3 + \dfrac{k_{17} \oplus k_{18} \oplus \cdots \oplus k_{24}}{256} \\ d_4 = c_4 + \dfrac{k_{25} \oplus k_{26} \oplus \cdots \oplus k_{32}}{256} \end{cases} \tag{2}$$

  where $\oplus$ represents bitwise XOR operation.
- Step 3: The initial values x_1 to x_6 of 6D hyper-chaotic system could be obtained from the 4 intermediate parameters by Equation (3).

$$\begin{cases} x_1 = \dfrac{((d_1 + d_2) \times 10^8)\ mod\ 256}{255} \\ x_2 = \dfrac{((d_2 + d_3) \times 10^8)\ mod\ 256}{255} \\ x_3 = \dfrac{((d_3 + d_4) \times 10^8)\ mod\ 256}{255} \\ x_4 = \dfrac{((d_1 + d_3) \times 10^8)\ mod\ 256}{255} \\ x_5 = \dfrac{((d_1 + d_4) \times 10^8)\ mod\ 256}{255} \\ x_6 = \dfrac{((d_2 + d_4) \times 10^8)\ mod\ 256}{255} \end{cases} \tag{3}$$

  where *mod* means module operation.

3.1.2. Hyper-Chaotic Matrices Generation

With the initial values that are calculated in Section 3.1.1, chaotic matrices could be generated from 6D hyper-chaotic system. In HCZRNA, chaotic matrices would be utilized in three parts, which are permutation, 3D transformed Zigzag diffusion, and RNA operation. Suppose that the plaintext image has $N \times N \times 3$ pixels, an $N \times N \times 6$ chaotic matrix is needed for permutation, a $2 \times N \times N \times 6$ chaotic matrix for Zigzag, and 64×6 chaotic matrix for RNA.

Therefore, the 6D hyper-chaotic system utilizes initial values from Equation (3) to iterate for generating a $(3 \times N \times N + 64) \times 6$ matrix. Given that i^{th} iteration's state values could be described as $s^i = \{x_{1,i}, x_{2,i}, x_{3,i}, x_{4,i}, x_{5,i}, x_{6,i}\}$, a hyper-chaotic matrix S could be depicted as Equation (4) after all iterations.

$$S = \{s^1, s^2, \ldots, s^M\} = \begin{Bmatrix} x_{1,1}, x_{1,2}, \ldots, x_{1,M} \\ x_{2,1}, x_{2,2}, \ldots, x_{2,M} \\ x_{3,1}, x_{3,2}, \ldots, x_{3,M} \\ x_{4,1}, x_{4,2}, \ldots, x_{4,M} \\ x_{5,1}, x_{5,2}, \ldots, x_{5,M} \\ x_{6,1}, x_{6,2}, \ldots, x_{6,M} \end{Bmatrix}_{6 \times M} \tag{4}$$

where $M = 3 \times N \times N + 64$.

However, the numbers in matrix S are double-precision values, which are suitable for permutation but not for Zigzag and RNA, and color image only has three channels

that are smaller than channels of S. Hence, matrix S should be separated into three pieces respectively.

For permutation, a matrix S_1 is calculated from the first $N \times N$ part of S by Equation (5).

$$S_1 = \begin{Bmatrix} x_{1,1} + x_{2,1}, x_{1,2} + x_{2,2}, \ldots, x_{1,M'} + x_{2,M'} \\ x_{3,1} + x_{4,1}, x_{3,2} + x_{4,2}, \ldots, x_{3,M'} + x_{4,M'} \\ x_{5,1} + x_{6,1}, x_{5,2} + x_{6,2}, \ldots, x_{5,M'} + x_{6,M'} \end{Bmatrix}_{3 \times M'} \tag{5}$$

where $M' = N \times N$.

While matrix S_2 is cut from $s^{M'+1}$ to $s^{3M'}$ in S for 3D transformed Zigzag diffusion. Additionally, because 8-bit integer digits are needed for diffusion, each item $x'_{i,j}$ in S_2 should be calculated by Equation (6).

$$\begin{aligned} Suppose \quad & \vec{s_i} = \{x_{i,(M'+1)}, x_{i,(M'+2)}, \ldots, x_{i,3M'}\} \\ & y_{i,j} = 2 \times x_{i,j} + \frac{max(\vec{s_i}) + min(\vec{s_i})}{max(\vec{s_i}) - min(\vec{s_i})} \\ & x'_{i,j} = ((\lfloor |y_{i,j}| - \lfloor |y_{i,j}| \rfloor \rfloor) \times 10^{10}) \ mod \ 256 \end{aligned} \tag{6}$$

where max and min are maximum and minimum operations.

Matrix S_3 is the last part of matrix S and it is used to sort operation for encrypting RNA codons tables as indexes. Because there only needs two indexes sequences, matrix S_3 should be summarized as Equation (7).

$$\begin{aligned} index_1 &= \{x_{1,3M'+1} + x_{2,3M'+1} + x_{3,3M'+1}, x_{1,3M'+2} + x_{2,3M'+2} + x_{3,3M'+2}, \ldots, x_{1,3M'+64} + x_{2,3M'+64} + x_{3,3M'+64}\} \\ index_2 &= \{x_{4,3M'+1} + x_{5,3M'+1} + x_{6,3M'+1}, x_{4,3M'+2} + x_{5,3M'+2} + x_{6,3M'+2}, \ldots, x_{4,3M'+64} + x_{5,3M'+64} + x_{6,3M'+64}\} \end{aligned} \tag{7}$$

3.1.3. Permutation

In this part, matrix S_1 is used to permute plaintext image. At the beginning, each element in S_1 should be allocated to each pixel as index. Hence, an $N \times N \times 3$ matrix S'_1 is needed to be converted from S_1 by reshaping.

$$S'_1 = reshape(S_1, N, N, 3) \tag{8}$$

Afterwards, each pixel in plaintext image has a corresponding index in S'_1 at the same coordinate. Combine plaintext image with matrix S'_1, and take another reshaping operation to convert these two matrix into two sequences with a length of $N \times N \times 3$. After sorting S'_1 ascendingly with image sequence synchronously, pixels' orders in plaintext image sequence have been scrambled.

Finally, reshaping the sorted image sequence to an $N \times N \times 3$ matrix, the initial cipher image could be generated.

3.1.4. Diffusion

After permutation, a diffusion scheme by 3D Zigzag transformation is proposed, as follows. An initial cipher image would be split and placed on the surfaces of an $N \times N \times 6$ cube, termed as P, as described in Section 2.2. Additionally, chaotic matrix S_2 would also be placed on another two $N \times N \times 6$ cubes, since diffusion would implement two rounds. For the first $N \times N \times 6$ numbers in S_2, each number would be placed on a cube in order, which could be called cube SC_1.For the last $N \times N \times 6$ numbers in S_2, cube SC_2 could be generated by the same process.

Subsequently, diffusion would start from origin point of cube P on the front side, and its coordinate is $[1,1,1]$. At each iteration, the pixel's value $C_{i,j,m}$ is calculated by Equation (9).

$$C_{i,j,m} = (P_{i,j,m} \oplus (T + x_{i,j,m;1}))\ mod\ 256 \tag{9}$$

where i, j, m are the coordinates of pixel at the i^{th} row, j^{th} column, and m^{th} side on the cube. T is the previous one diffused pixel's C value, if $i, j, m = 1, 1, 1$, T is a user-defined constant. $x_{i,j,m;1}$ is the corresponding coordinate's value in SC_1.

For the second round of diffusion, Equation (9) would change to Equation (10).

$$D_{i,j,m} = (C_{i,j,m} \oplus (T' + x_{i,j,m;2}))\ mod\ 256 \tag{10}$$

where D is result of diffusion, and T' is the previous one diffused pixel's D value, and, if $i, j, m = 1, 1, 1$, T' is the last pixel's C value after the first round diffusion. While $x_{i,j,m;2}$ is corresponding coordinate's value in SC_2.

Through these two round diffusions, D cube is generated. Additionally, recover the D's $N \times N \times 6$ matrix by reversing processes of image splitting and cube placement in Section 2.2. A diffused $N \times N \times 3$ matrix D_{mat} is obtained.

3.1.5. RNA Operation

The encryption from diffused matrix D_{mat} through RNA operation could be described, as follows:

- Step 1: RNA operation is initiated from creating two encrypted codons tables, called T_{00} and T_{01}. In which, T_{00} and T_{01} are shuffled tables from codons truth, as in Table 2. The shuffle orders are generated according to indexes sequences calculated from Equation (7). After sorting with these two indexes sequences, the original codons truth table could be shuffled to two different encrypted codons tables T_{00} and T_{01}. Subsequently, by the complementary rules of RNA, additional tables T_{10} and T_{11} could be generated from T_{00} and T_{01}. Hence, four encrypted codons tables are generated.
- Step 2: for each element in D_{mat}, binary number conversion is processed, which is recorded as B.

$$B = \{b_{i,j,m}\}. \quad i, j = 1, 2, \ldots, N; m = 1, 2, 3 \tag{11}$$

  Each $b_{i,j,m}$ could be expressed as eight binary numbers, which could be depicted as $b_0^{i,j,m} b_1^{i,j,m} b_2^{i,j,m} b_3^{i,j,m} b_4^{i,j,m} b_5^{i,j,m} b_6^{i,j,m} b_7^{i,j,m}$.
- Step 3: divide $b_{i,j,m}$ into four pieces, each two bits are one piece, which are recorded as:

$$\begin{aligned} bt_1^{i,j,m} &= b_0^{i,j,m} b_1^{i,j,m} \\ bt_2^{i,j,m} &= b_2^{i,j,m} b_3^{i,j,m} \\ bt_3^{i,j,m} &= b_4^{i,j,m} b_5^{i,j,m} \\ bt_4^{i,j,m} &= b_6^{i,j,m} b_7^{i,j,m} \end{aligned} \tag{12}$$

  Additionally, combine three channels' bts at the same coordinate together:

$$\begin{aligned} bt_1^{i,j} &= bt_1^{i,j,1} bt_1^{i,j,2} bt_1^{i,j,3} \\ bt_2^{i,j} &= bt_2^{i,j,1} bt_2^{i,j,2} bt_2^{i,j,3} \\ bt_3^{i,j} &= bt_3^{i,j,1} bt_3^{i,j,2} bt_3^{i,j,3} \\ bt_4^{i,j} &= bt_4^{i,j,1} bt_4^{i,j,2} bt_4^{i,j,3} \end{aligned} \tag{13}$$

  Therefore, each $bt^{i,j}$ has six bits that could transfer to RNA codons according to Table 1. Exchange each two bits in bts to RNA base one-by-one according to the principle

of row priority, *bt*s could be coded to codons. And put them into a one-dimension sequence BS as Equation (14).

$$BS = \{bt_1^{1,1}, bt_2^{1,1}, bt_3^{1,1}, bt_4^{1,1}, bt_1^{1,2}, bt_2^{1,2}, \ldots, bt_4^{2,1}, bt_1^{2,2}, \ldots, bt_3^{N,N}, bt_4^{N,N}\}. \quad (14)$$

- Step 4: convert key to binary format. 256-bit key could be changed into a binary sequence BK.

$$\begin{aligned} key &= [key_0, key_1, \ldots, key_{31}] \\ key_i &= key_{i,0}, key_{i,1}, \ldots, key_{i,7} \\ BK &= [key_{1,0}, key_{1,1}, \ldots, key_{1,7}, key_{2,0}, key_{2,1}, \ldots, key_{31,7}] \end{aligned} \quad (15)$$

Walk through sequence BS, and find corresponding index id of each codon in BS from Table 2. For each codon in BS, check 2-bits table number z in sequence BK.

$$z = BK_{n \ mod \ 2048} BK_{(n+1) \ mod \ 2048} \quad (16)$$

where n is the walking times.
Take the codon $T_z(id)$ to replace the origin codon $BS(n)$.
When iterations termination, an encrypted sequence is generated.

- Step 5: decode each base in encrypted sequence BS to binary format by Table 1, put all of the binary digits back to original coordinates by reversing operations in Step 3. Additionally, change binary matrix into 2-bit matrix. The cipher image is generated.

The HCZRNA encryption has four stages: hyper-chaotic matrices generation (Sections 3.1.1 and 3.1.2), hyper-chaotic permutation (Section 3.1.3), 3D transformed Zigzag diffusion on surfaces of cubes, which is generated from initial cipher image (Section 3.1.4), and a bit-level RNA operation (Section 3.1.5). The major steps of the HCZRNA are Sections 3.1.4 and 3.1.5, i.e., the transformed Zigzag diffusion on 3D cubes and bit-level RNA substitutions with hyper-chaotic matrix, respectively. The HCZRNA uses the strategy of "divide and conquer" that is widely used in various applications to decompose the original encryption task into a couple of simpler sub-tasks [38,39].

3.2. Decryption

In this paper, the encryption scheme has been depicted, and decryption is the inverse process of encryption. Details are proposed, as follows.

- Step 1: redo the processes that are listed in Sections 3.1.1 and 3.1.2 to generate hyper-chaotic matrices S_1, S_2, and S_3.
- Step 2: convert the cipher image to a binary format, and reconstruct three channels' pixels at each coordinate into four 6-bit binary arrays by using Euqation (12) and (13). Change 6-bit arrays into codons from codons truth Table 2, and put them in a one-dimension sequence BS' as Equation (14).
- Step 3: generate key binary sequence BK through Equation (15) and encrypted codons tables $\{T_{00}, T_{01}, T_{10}, T_{11}\}$ by redoing Step 1 in Section 3.1.5.
- Step 4: Check each 2-bits z in BK and find corresponding table T_z from $\{T_{00}, T_{01}, T_{10}, T_{11}\}$. Walk through BS' and find each codon's corresponding index id' in Table T_z. Replace codon in BS' to codon id' in codons truth Table 2. After all codons are replaced, convert them into binary formats and 8-bit numbers, matrix D'_{mat} is obtained.
- Step 5: split matrix D'_{mat} and place triangles on cube surfaces as the process shown in Section 2.2. Redo Section 3.1.4 with modified Equation (17) two rounds, and then walk through pixels with reversed Zigzag path. Take Figure 4 in Section 2.2 as an

example, the traversal road of decryption is shown in Figure 7. If we put all the pixels together, the order of traversal is depicted in Figure 8.

$$\begin{aligned} C'_{i,j,m} &= (D'_{i,j,m} \oplus (T' + x_{i,j,m;2}))mod256 \\ P'_{i,j,m} &= (C'_{i,j,m} \oplus (T + x_{i,j,m;1}))mod256 \end{aligned} \tag{17}$$

where T' is the previous one pixel's D' value, and, if $i, j, m = 1, 1, 1$, T' is the last pixel's C' value after first round iteration. T is the previous one pixel's C' value and, if $i, j, m = 1, 1, 1$, T is the user-defined constant that is used in Section 3.1.4.

- Step 6: after the process in Step 5, return the triangles in the cube to theirs original coordinates on a image. Additionally, the reverse processes in Section 3.1.3, reshape S_1 to construct sorted sequence. Find image pixels' corresponding coordinates through sorted sequence and recover. The decrypted image is generated.

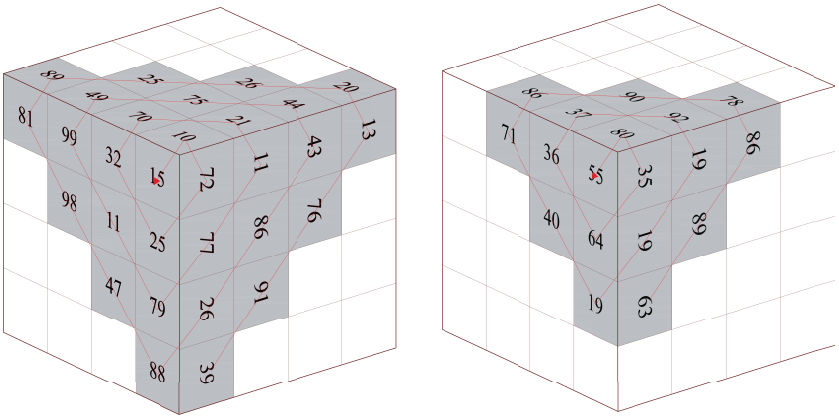


Figure 7. Reverse traversal.

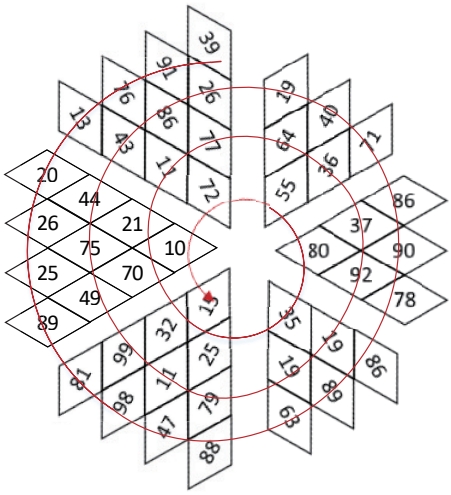


Figure 8. The order of Zigzag in decryption.

4. Experimental Results

The encryption and decryption schemes have been tested on four popular RGB color images in Table 3. All of the experiments are conducted by MATLAB R2019b on 64-bit Windows 10 system, and the main hardware includes an Xeon(R) W-2223 @ 3.60 GHz CPU as well as 32 GB RAM.

Table 3. Testing images.

Image	Size ($h \times w \times c$)	Image	Size ($h \times w \times c$)
Lena	$256 \times 256 \times 3$	Baboon	$512 \times 512 \times 3$
Peppers	$256 \times 256 \times 3$	Splash	$512 \times 512 \times 3$

For the controlling parameters setting in Equation (1), $(a, b, c, d, e, f, g, r, \omega, \beta) = (0.3, 1.5, 8.5, -2, 1, -0.1, 0.9, 1, 1, 0.2)$. Constants c_1, c_2, c_3, c_4 in Equation (2) are set as $(1, 1, 2, 2)$ and the initial constant of T in Equation (9) is 11. The security key can be set by users, so we set a 256-bit hexadecimal sequence that is shown below as the security key in all of the experiments. The key can also be optimized by some evolutionary optimizations, such as differential evolution and particle swarm optimization [40–43].

$$key =' 743B5A203B1E8EDF6C0FB0D7497CB2E228689AD00F57F8953B5C6127E1C26053'$$

In order to demonstrate performance of proposed HCZRNA scheme, five state-of-the-art encryption schemes are employed for comparison: a Four-wing hyper-chaotic system based dynamic DNA encryption scheme [29], an extended Zigzag confusion and RNA encryption based scheme [23], a Hopfield chaotic neural network-based scheme [44], a scheme with utilization of differences between two 1D chaotic maps [45] and a scheme with 4D hyper-chaotic system and DNA encryption [46].

4.1. Key Space

For an image encryption system, large enough key space is necessary to withstand a brute-force attack. In HCZRNA, a 256-bit security key is used to calculate the initial values of the hyper-chaotic system to generate the pseudo random matrices that could affect the outputs of permutation, diffusion, and RNA operations. As we know, different initial values in a hyper-chaotic system would get different pseudo random sequences, and each bit has two states, the security key has 2^{256} different states, so it could generate 2^{256} results of a hyper-chaotic system. Therefore, the key space of HCZRNA could be calculated as 2^{256}. Theoretically, if the key space of an encryption scheme is larger than 2^{100}, this scheme could resist violent crack by modern computers [47]. Therefore, the proposed HCZRNA in this paper has a large enough key space to resist brute-force attack.

4.2. Sensitivity of Keys

The sensitivity test on keys refers to utilize slightly different keys to encrypt the same images. If an encryption is sensitive, the encryption with slight difference on keys would get completely different cipher images. To test the key sensitivity, we would use two different keys to encrypt four test images, one of these two keys is initial security key key_1, another key is key_2, which is one bit changed for key_1. These two keys are stated as follows, where the changed bits are shown in red:

$$key_1 =' 743B5A203B1E8EDF6C0FB0D7497CB2E228689AD00F57F8953B5C6127E1C26053'$$
$$key_2 =' 743B5A203B1F8EDF6C0FB0D7497CB2E228689AD00F57F8953B5C6127E1C26053'$$

By comparing two cipher images from the same plaintext image, the differences of cipher images that are encrypted from these two security keys are stated in Table 4.

Table 4. Differences between the cipher images.

Image	Lena	Pepper	Baboon	Splash
Difference	99.59%	99.62%	99.61%	99.60%

In the table, it is obvious that all of the differences between two cipher images are over 99%, which reveals that, even with tiny changes in security keys, encryption by HCZRNA

would also lead to extremely different outputs. Hence, HCZRNA satisfies sensitivity requirements.

4.3. Histogram

Because a histogram reflects each pixel's times in an image, histograms of meaningful images are fluctuated, while cipher images' histogram should be flat and uniform. That is to say, if an encryption scheme is well-designed, the histograms of cipher images should be as flat as possible. For the proposed HCZRNA, a histogram of Baboon and its cipher image are placed in Figure 9.

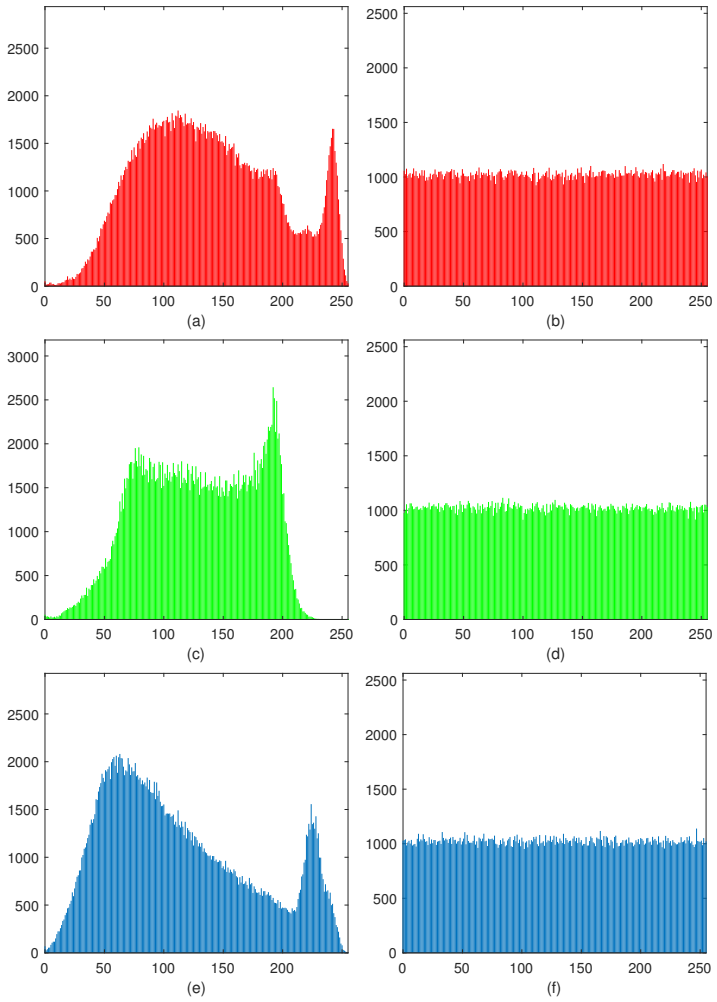


Figure 9. Histogram. image (**a**,**c**,**e**) are the histograms of three channels of Baboon, and image (**b**,**d**,**f**) are the histograms of corresponding channels of encrypted Baboon.

From this figure, it could find that histograms of all channels in plaintext image are fluctuated, while histograms of cipher image's different channels are almost distributed in a narrow range, and their values are around 1000. For more accurate results, histogram statistics are introduced to evaluate the variance and standard deviation of plaintext and cipher images [48,49]. Variance is used to calculate the average difference in each gray level frequency with respect to mean value $\bar{x}$, which could be formulated as Equation (18).

$$\begin{aligned} \alpha &= \frac{1}{256}\sum_{i=1}^{256}(x_i - \bar{x})^2, \\ \bar{x} &= \frac{h \times w}{256} \end{aligned} \tag{18}$$

where h, w represent the image's height and width respectively, x is the frequency of different gray levels of pixels in a image, and the $\bar{x}$ is the mean value of xs. And α is the variance, the higher is α, the more fluctuate is the graphic histogram. Accordingly, if a encryption is well-designed, the α of encrypted image should be low.

As α is always very high in plaintext image, a standard deviation is used to evaluate histogram's fluctuations, which is stated as Equation (19).

$$\beta = \sqrt{\alpha} \tag{19}$$

where β is the standard deviation. For all test images, Table 5 describes the results of histogram statistics.

Table 5. Histogram statistics.

Image	Channels	Plaintext		Ciphertext	
		α	β	α	β
Lena	R	65,306	255	248	15
	G	30,665	175	258	16
	B	91,939	303	232	15
Pepper	R	57,413	239	249	15
	G	119,411	345	238	15
	B	151,644	389	237	15
Baboon	R	165,679	407	520	22
	G	285,616	534	532	23
	B	159,885	399	541	23
Splash	R	1,211,325	1100	566	23
	G	1,541,948	1241	495	22
	B	2,958,482	1720	504	22

In the table, the variances and standard deviations of plaintext images are very high, while they are extremely different in cipher images. All of these performances indicate that the proposed HCZRNA could effectively resist histogram attack.

4.4. Correlation

The correlation test refers to adjacent pixels' relationship. A meaningful image has high correlation because values of adjacent pixels are close to each other. This attribute could be utilized to crack. Therefore, a well-designed encryption scheme should have low enough correlations in three directions: horizontal, vertical, and diagonal directions. Given a pixel sequence that is represented by $X = \{x_1, x_2, \ldots, x_N\}$ and its adjacent pixel sequence $Y = \{y_1, y_2, \ldots, y_N\}$ in an image, correlation between X and Y could be denoted as $\gamma_{X,Y}$ in Equation (20).

$$\begin{aligned} \gamma_{X,Y} &= \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - D(X))(y_i - D(Y))}{\sqrt{D(X)D(Y)}} \\ D(X) &= \frac{\sum_{i=1}^{N}(x_i - E(X))^2}{N} \\ E(X) &= \frac{\sum_{i=1}^{N} x_i}{N} \end{aligned} \tag{20}$$

where $E(X)$ is X's mathematical expectation and $D(X)$ is standard deviation.

If X and Y are identical, $\gamma_{X,Y}$ would be a maximum of 1. On the contrary, $\gamma_{X,Y}$ would be close to 0 when X and Y have few correlations.

Figure 10 depicts the correlation test results. It is obvious that the adjacent pixels' distributions in plaintext images are concentrated, while the distributions in the cipher images are opposite.

More accurately, Table 6 provides correlation coefficients between plaintext images and cipher images. Additionally Table 7 demonstrates comparisons with references [44,45]. Through this test, it could find that the correlation coefficients of the proposed HCZRNA are extremely close to 0, which means that HCZRNA could effectively break correlations existing in plaintext images. While the comparisons show that the proposed HCZRNA achieves the best results with [44,45] in all cases. This reveals that HCZRNA outperforms when compared schemes in terms of reducing correlations.

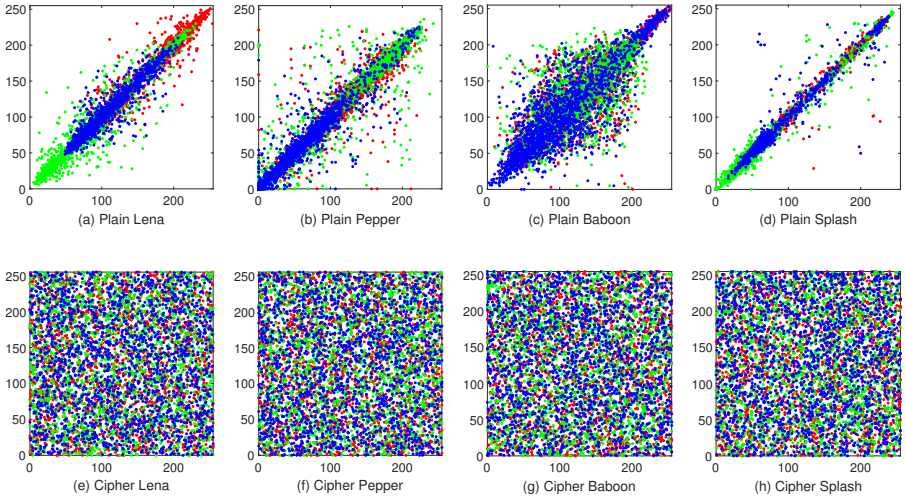


Figure 10. Correlations.The first row is correlations of plaintext images, and the second row is correlations of cipher images.

Table 6. The correlation coefficients of the testing images.

Image	Channels	Plaintext			Ciphertext		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	R	0.9512	0.9755	0.9444	0.0046	0.0024	0.0051
	G	0.9512	0.9679	0.9276	−0.0027	−0.0007	0.0002
	B	0.9512	0.9479	0.9021	−0.0023	0.0014	0.0004
Baboon	R	0.9218	0.8624	0.8531	0.0003	0.0001	0.0015
	G	0.9218	0.7591	0.7299	−0.0010	0.0004	0.0020
	B	0.9218	0.8782	0.8411	0.0005	−0.0022	0.0012

Table 7. Comparisons of correlation coefficients.

Image	Channels		Plaintext	Ciphertext		
				HCZRNA	Ref. [44]	Ref. [45]
Baboon	R	Horizontal	0.9218	**0.0003**	0.0054	−0.0073
		Vertical	0.8624	**0.0001**	−0.0042	−0.0059
		Diagonal	0.8531	**0.0015**	−0.0177	−0.0136
	G	Horizontal	0.9218	**−0.0010**	−0.0055	0.0046
		Vertical	0.7591	**0.0004**	0.0119	−0.0077
		Diagonal	0.7299	**0.0020**	0.0046	−0.0044
	B	Horizontal	0.9218	**0.0005**	−0.0021	−0.0067
		Vertical	0.8782	**−0.0022**	0.0104	−0.0111
		Diagonal	0.8411	**0.0012**	−0.0021	0.0122

4.5. Information Entropy

Information Entropy shows the randomness and uncertainty of image's pixels. If pixels in an image have uniform distribution, this image could resistant statistical attacks. Because there are 256 gray levels in each channel of color image, the Entropy calculation could be formulated as Equation (21):

$$H(C) = -\sum_{i=0}^{255} p(i) log_2 p(i) \tag{21}$$

where C denotes channels of color image and $p(i)$ is probability of gray level in whole channel.

The bigger $H(C)$, the bigger uncertainty of image. While the theoretical value of $H(C)$ is 8.

Table 8 shows the entropies of all channels of plaintext color images and corresponding cipher images through encryptions by proposed HCZRNA. It is obvious that cipher images have increased entropies a lot from plaintext images and their entropies are very close to the theoretical value. Moreover, a comparison is held between HCZRNA and Ref. [23,29,44–46], and the results are stated in Table 9. Among all of the encryption schemes, the proposed HCZRNA achieves the highest entropies in four out of six cases. It could conclude that HCZRNA has the ability to resist statistical attack.

Table 8. Information Entropies of testing images.

Image	Channels	Lena	Peppers	Baboon	Splash
Plaintext	R	7.2353	7.3369	7.7067	6.9481
	G	7.5683	7.4394	7.4744	6.8845
	B	6.9176	7.0219	7.7522	6.1265
Ciphertext	R	7.9973	7.9972	7.9993	7.9993
	G	7.9970	7.9970	7.9993	7.9994
	B	7.9972	7.9972	7.9993	7.9993

Table 9. Comparison of entropies.

Image	Channel	Plaintext	HCZRNA	Ref. [29]	Ref. [23]	Ref. [44]	Ref. [45]	Ref. [46]
Lena	R	7.2353	**7.9973**	7.9971	**7.9973**	-	-	**7.9973**
	G	7.5683	7.9970	7.9971	7.9972	-	-	**7.9975**
	B	6.9176	7.9972	7.9971	7.9971	-	-	**7.9975**
Baboon	R	7.7067	**7.9993**	7.9926	-	**7.9993**	**7.9993**	7.9970
	G	7.4744	**7.9993**	7.9926	-	**7.9993**	**7.9993**	7.9978
	B	7.7522	**7.9993**	7.9926	-	**7.9993**	7.9992	7.9987

4.6. Differential Attack

The differential attack test is an important security test for image encryption, which reveals the influence on the cipher image caused by a minor change in pixels of plaintext image. If a tiny change on pixels in plaintext image leads to significant different cipher image, that is to say the encryption scheme could resist differential attack.

Two important indices are introduced to measure the ability of differential attack resistance, which is called the number of pixel change rate (NPCR) and the unified average changing intensity (UACI). Additionally, they are defined as Equations (22) and (23):

$$NPCR = \frac{\sum_{i=0}^{h} \sum_{j=0}^{w} F(i,j) \times 100\%}{w \times h} \tag{22}$$

$$UACI = \frac{\sum_{i=0}^{h}\sum_{j=0}^{w}|e_1(i,j) - e_2(i,j)|}{255 \times w \times h} \tag{23}$$

where e_1 and e_2 are two cipher images, and $e(i,j)$ means the pixel's value at coordinate i,j in image e. $F(i,j)$ denotes whether the same coordinate's pixel values in e_1 and e_2 are independent or not, which could be formulated as Equation (24):

$$F(i,j) = \begin{cases} 0, & if \quad e_1(i,j) = e_2(i,j) \\ 1, & if \quad e_1(i,j) \neq e_2(i,j) \end{cases} \tag{24}$$

For two random images, NPCR and UACI's expected values are stated as: $NPCR = 99.6094\%$ and $UACI = 33.4635\%$ for an 8-bit gray image [30].

Hence, to realize the test, one bit would be changed on a random pixel in plaintext image. And both the plaintext image and changed image are encrypted to two different cipher images. Table 10 lists the average results of ten times tests. It could find that all NPCR values and UACI values of cipher images' different channels exceed the theoretical values. Additionally, comparisons with Ref. [23,29,44–46] are shown in Tables 11 and 12. Through the comparisons, the proposed HCZRNA encryption scheme has better performances on NPCR and UACI, which indicates that HCZRNA could resist differential attack well.

Table 10. The mean number of pixel change rate (NPCR) and unified average changing intensity (UACI) of cipher images.

Image	NPCR(%)			UACI(%)		
	R	G	B	R	G	B
Lena	99.6619	99.6272	99.6460	33.6177	33.6048	33.6422
Peppers	99.6481	99.6404	99.6239	33.7208	33.5701	33.6435
Baboon	99.6159	99.6769	99.6115	33.5196	33.5203	33.5049
Splash	99.6219	99.6934	99.6253	33.4983	33.5114	33.4816

Table 11. Average NPCR (%) of running the schemes 10 times.

Image	Channel	HCZRNA	Ref. [29]	Ref. [23]	Ref. [44]	Ref. [45]	Ref. [46]
Lena	R	**99.6619**	99.60	99.6323	-	-	99.615
	G	**99.6272**	*99.61*	99.6109	-	-	99.62
	B	**99.6460**	*99.61*	99.6338	-	-	99.617
Baboon	R	**99.6159**	99.6083	-	*99.6037*	*99.6037*	99.6140
	G	**99.6769**	99.6065	-	99.6048	*99.6017*	99.6073
	B	99.6115	99.6094	-	99.6059	*99.6043*	**99.6292**

Table 12. Average UACI (%) of running the schemes 10 times.

Image	Channel	HCZRNA	Ref. [29]	Ref. [23]	Ref. [44]	Ref. [45]	Ref. [46]
Lena	R	**33.6177**	33.56	*33.4683*	-	-	33.4732
	G	**33.6048**	33.45	*33.4341*	-	-	33.3428
	B	**33.6422**	33.49	33.4991	-	-	*33.4647*
Baboon	R	**33.5196**	33.4939	-	33.4427	*29.9630*	33.4843
	G	**33.5203**	33.4295	-	33.4605	*28.5708*	33.4690
	B	**33.5049**	33.4856	-	31.9747	*31.2574*	33.4965

4.7. Robustness

It is unavoidable that there data loss or noise attack occur when cipher images are transmitting. Hence, a well-designed encryption and decryption scheme should resist contamination on cipher images to recover plaintext images without great changes.

To demonstrate robustness of proposed HCZRNA scheme, 12.5%, 25%, and 50% data lose tests and 1%, 5%, and 10% salt and pepper noise tests would presented in Figures 11 and 12.

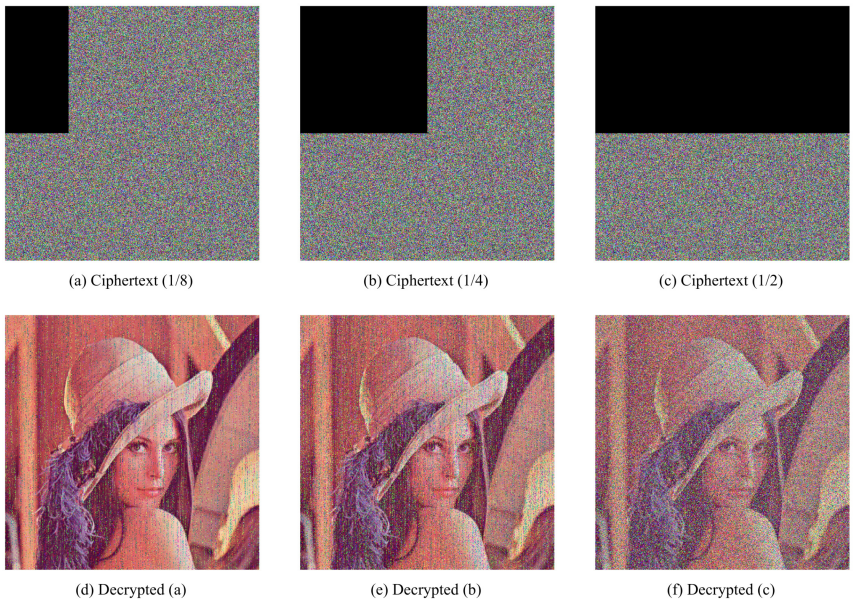


Figure 11. Cropping attack tests. The first row is cipher images with 12.5%, 25% and 50% data loss, and the second row is decrypted images from the first row.

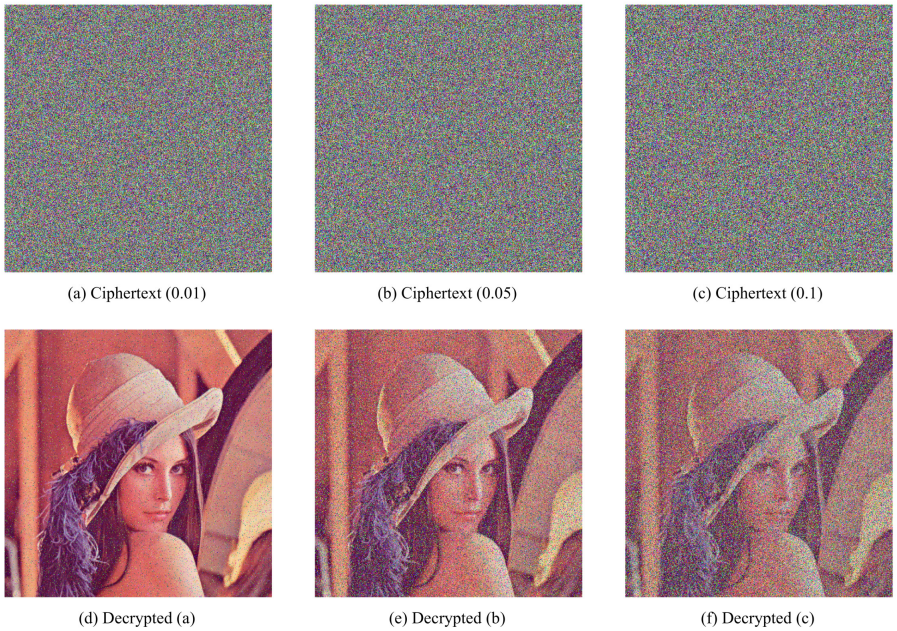


Figure 12. Noise attack tests. The first row is cipher images with 1%, 5%, 10% salt and pepper noise, and the second row is decrypted images from the first row.

From the figures, the main information of plaintext images could be identified from decrypted images, which could conclude that HCZRNA has enough robustness for data loss and noise attacks. Here, the Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) are also utilized to test robustness [48,49], which is formulated as Equation (25).

$$MSE = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} [P(i,j) - E(i,j)]^2,$$
$$PSNR = 20log_{10}(\frac{255}{\sqrt{MSE}}) \tag{25}$$

where P and E represent two different images. MSE is used to evaluate the difference between two images, and PSNR depicts the ratio between the maximum possible power of a signal and the power of distorting noise that affects the quality of its representation. The lower the MSE, the higher PSNR, which indicates that two images have high similarity. Hence, under noise attacks, if the PSNR between the plaintext image and decrypted image is high, the encryption and decryption schemes are good enough. Tables 13 and 14 present the results of plaintext image and decrypted image of Lena under data loss and salt and pepper noise attacks.

Table 13. Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) under data loss.

Data Loss	MSE	PSNR
12.5%	1195	17.35
25%	2315	14.48
50%	4502	11.60

Table 14. MSE and PSNR under salt and pepper noise.

Salt and Pepper Noise	MSE	PSNR
1%	610	20.28
5%	2684	13.84
10%	4490	11.61

Through the results, we could find there are high MSEs and low PSNRs in these tables, which figures out that HCZRNA could resist attacks of data loss and noise.

4.8. Running Time

In HCZRNA, the pixels of image would be walked through multiple times in diffusion and RNA operation. Suppose that the size of RGB image is $N \times N \times 3$. For the five parts of encryption processes that are listed in Section 3.1, initial values of hyper-chaotic system are calculated from the security key, which costs $O(1)$ time complexity; the hyper-chaotic matrices are computed $3 \times N \times N + 64$ times iterations; for permutation, reshape and sort operations are implemented three times; while the diffusion process walks through each pixel two times, which costs $O(2 \times N \times N \times 3)$; at last, as RNA operation walks through all 6-bit codons that are transformed from 8-bit pixels, the times of iteration are increased to $\frac{4}{3} \times N \times N \times 3$. Hence, the time complexity of HCZRNA could be calculated as $O(1 + 3 \times N \times N + 64 + 3 + 2 \times N \times N \times 3 + \frac{4}{3} \times N \times N \times 3) = O(13N^2 + 68) = O(N^2)$. Using the experiment environment that is listed in this section, the running times of encryption and decryption could be stated in Table 15. Although the time costs of encryption and decryption are not very good, the time complexity is also a polynomial time, which could be tolerable. Additionally, the processes of RNA operation on different codons have no correlation with each other, which could improve computational time by computing RNA operation in parallel.

Table 15. Running time (unit: second).

Image Size	Encryption	Decryption
$64 \times 64 \times 3$	0.59	0.44
$128 \times 128 \times 3$	2.39	1.77
$256 \times 256 \times 3$	9.36	6.96
$512 \times 512 \times 3$	37.54	28.49

5. Conclusions

A novel hyper-chaotic system based image encryption scheme is proposed with 3D transformed Zigzag and RNA operation in this paper. By using the 6D hyper-chaotic system, three auxiliary matrices are generated, including one permutation index matrix, one mask matrix for Zigzag, and one codon table index matrix. Subsequently, two rounds 3D transformed Zigzag diffusion mechanism is proposed for pixels diffusion with each other. Nevertheless, additional encryption with RNA codons makes more reliable and secure results through employing codons tables and security keys. Through simulations, the proposed HCZRNA has better performances on the resistance of different types attacks than the compared encryption schemes, while the speed is not ideal, since it is a complex process. On the premise of ensuring performance, we would simplify diffusion and RNA operation processes and optimize the encryption steps for improving speed in the future.

Author Contributions: Conceptualization, D.Z. and T.L.; Data curation, L.C.; Formal analysis, D.Z.; Funding acquisition, T.L.; Investigation, T.L.; Methodology, D.Z. and T.L.; Project administration, T.L.; Resources, D.Z.; Software, D.Z.; Supervision, D.Z. and T.L.; Validation, L.C. and T.L.; Visualization, L.C.; Writing—original draft, D.Z.; Writing—review & editing, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Education of Humanities and Social Science Project (Grant No. 19YJAZH047) and the Scientific Research Fund of Sichuan Provincial Education Department (Grant No. 17ZB0433).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request.

Acknowledgments: We gratefully acknowledge the reviewers for their comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

6D	6 Dimensional
RNA	Ribonucleic acid
HCZRNA	Hyper-chaotic color image encryption mechanism based on transformed Zigzag diffusion and RNA operations
RGB	Red Greeen Blue

References

1. Sneha, P.S.; Sankar, S.; Kumar, A.S. A chaotic colour image encryption scheme combining Walsh–Hadamard transform and Arnold–Tent maps. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 1289–1308. [CrossRef]
2. Wang, H.; Xiao, D.; Chen, X.; Huang, H. Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map. *Signal Process.* **2018**, *144*, 444–452. [CrossRef]
3. Jeng, F.G.; Huang, W.L.; Chen, T.H. Cryptanalysis and improvement of two hyper-chaos-based image encryption schemes. *Signal Process. Image Commun.* **2015**, *34*, 45–51. [CrossRef]
4. Jithin, K.C.; Sankar, S. Colour image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set. *J. Inf. Secur. Appl.* **2020**, *50*, 102428. [CrossRef]
5. Zhang, Y. Test and verification of AES used for image encryption. *3D Res.* **2018**, *9*, 1–27. [CrossRef]

6. Xian, Z.H.; Sun, S.L. Image Encryption Algorithm Based on Chaos and S-Boxes Scrambling. *Adv. Mater. Res.* **2010**, *171–172*, 299–304. [CrossRef]
7. Zhou, Y.; Hua, Z.; Pun, C.M.; Chen, C.L.P. Cascade Chaotic System With Applications. *IEEE Trans. Cybern.* **2015**, *45*, 2001–2012. [CrossRef] [PubMed]
8. Liu, L.; Wang, Y.N.; Hou, L.; Feng, X.R. Easy encoding and low bit–error–rate chaos communication system based on reverse–time chaotic oscillator. *IET Signal Process.* **2017**, *11*, 869–876. [CrossRef]
9. Zhang, L.; Liao, X.; Wang, X. An image encryption approach based on chaotic maps. *Chaos Solitons Fractals* **2005**, *24*, 759–765. [CrossRef]
10. Bouslehi, H.; Seddik, H. Innovative image encryption scheme based on a new rapid hyperchaotic system and random iterative permutation. *Multimed. Tools Appl.* **2018**, *77*, 30841–30863. [CrossRef]
11. Zhang, Y.; Wen, W.; Su, M.; Li, M. Cryptanalyzing a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Optik* **2014**, *125*, 1562–1564. [CrossRef]
12. Mohammad Seyedzadeh, S.; Mirzakuchaki, S. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Process.* **2012**, *92*, 1202–1215. [CrossRef]
13. Li, T.; Shi, J.; Zhang, D. Color image encryption based on joint permutation and diffusion. *J. Electron. Imaging* **2021**, *30*, 013008. [CrossRef]
14. Askar, S.S.; Karawia, A.A.; Alshamrani, A. Image Encryption Algorithm Based on Chaotic Economic Model. *Math. Probl. Eng.* **2015**, *2015*, 1–10. [CrossRef]
15. Shaikh, N.; Chapaneri, S.; Jayaswal, D. Hyper chaotic color image cryptosystem. In Proceedings of the 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Coimbatore, India, 24 October 2016; pp. 239–243. [CrossRef]
16. Li, C.; Zhao, F.; Liu, C.; Lei, L.; Zhang, J. A Hyperchaotic Color Image Encryption Algorithm and Security Analysis. *Secur. Commun. Netw.* **2019**, *2019*, 1–8. [CrossRef]
17. Zhou, Y.; Bao, L.; Chen, C.P. A new 1D chaotic system for image encryption. *Signal Process.* **2014**, *97*, 172–182. [CrossRef]
18. Kadir, A.; Aili, M.; Sattar, M. Color image encryption scheme using coupled hyper chaotic system with multiple impulse injections. *Opt. Int. J. Light Electron Opt.* **2017**, *129*, 231–238. [CrossRef]
19. Li, C.; Zhang, L.Y.; Ou, R.; Wong, K.W.; Shu, S. Breaking a novel colour image encryption algorithm based on chaos. *Nonlinear Dyn.* **2012**, *70*, 2383–2388. [CrossRef]
20. Xingyuan, W.; Junjian, Z.; Guanghui, C. An image encryption algorithm based on ZigZag transform and LL compound chaotic system. *Opt. Laser Technol.* **2019**, *119*, 105581. [CrossRef]
21. Xu, X.; Feng, J. Research and Implementation of Image Encryption Algorithm Based on Zigzag Transformation and Inner Product Polarization Vector. In Proceedings of the 2010 IEEE International Conference on Granular Computing, San Jose, CA, USA, 14–16 August 2010; pp. 556–561. [CrossRef]
22. Li, Y.; Li, X.; Jin, X.; Zhao, G.; Ge, S.; Tian, Y.; Zhang, X.; Zhang, K.; Wang, Z. An Image Encryption Algorithm Based on Zigzag Transformation and 3-Dimension Chaotic Logistic Map. In *Applications and Techniques in Information Security*; Niu, W., Li, G., Liu, J., Tan, J., Guo, L., Han, Z., Batten, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 557, pp. 3–13. doi:10.1007/978-3-662-48683-2{\textunderscore }1. [CrossRef]
23. Wang, X.; Guan, N. A novel chaotic image encryption algorithm based on extended Zigzag confusion and RNA operation. *Opt. Laser Technol.* **2020**, *131*, 106366. [CrossRef]
24. Feixiang, Z.; Mingzhe, L.; Kun, W.; Hong, Z. Color image encryption via Hénon-zigzag map and chaotic restricted Boltzmann machine over Blockchain. *Opt. Laser Technol.* **2021**, *135*, 106610. [CrossRef]
25. Sahasrabuddhe, A.; Laiphrakpam, D.S. Multiple images encryption based on 3D scrambling and hyper-chaotic system. *Inf. Sci.* **2021**, *550*, 252–267. [CrossRef]
26. Hu, T.; Liu, Y.; Gong, L.H.; Ouyang, C.J. An image encryption scheme combining chaos with cycle operation for DNA sequences. *Nonlinear Dyn.* **2017**, *87*, 51–66. [CrossRef]
27. Li, T.; Yang, M.; Wu, J.; Jing, X. A novel image encryption algorithm based on a fractional-order hyperchaotic system and DNA computing. *Complexity* **2017**, *2017*, 9010251. [CrossRef]
28. Liu, Y.; Wang, J.; Fan, J.; Gong, L. Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences. *Multimed. Tools Appl.* **2016**, *75*, 4363–4382. [CrossRef]
29. Chai, X.; Fu, X.; Gan, Z.; Lu, Y.; Chen, Y. A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process.* **2019**, *155*, 44–62. [CrossRef]
30. Hu, T.; Liu, Y.; Gong, L.H.; Guo, S.F.; Yuan, H.M. Chaotic image cryptosystem using DNA deletion and DNA insertion. *Signal Process.* **2017**, *134*, 234–243. [CrossRef]
31. Wu, J.; Shi, J.; Li, T. A novel image encryption approach based on a hyperchaotic system, pixel-level filtering with variable kernels, and DNA-level diffusion. *Entropy* **2020**, *22*, 5. [CrossRef] [PubMed]
32. Liu, P.; Zhang, T.; Li, X. A new color image encryption algorithm based on DNA and spatial chaotic map. *Multimed. Tools Appl.* **2019**, *78*, 14823–14835. [CrossRef]
33. Mahmud, M.; ur Rahman, A.; Lee, M.; Choi, J.Y. Evolutionary-based image encryption using RNA codons truth table. *Opt. Laser Technol.* **2020**, *121*, 105818. [CrossRef]

34. Abbasi, A.A.; Mazinani, M.; Hosseini, R. Chaotic evolutionary-based image encryption using RNA codons and amino acid truth table. *Opt. Laser Technol.* **2020**, *132*, 106465. [CrossRef]
35. Yadollahi, M.; Enayatifar, R.; Nematzadeh, H.; Lee, M.; Choi, J.Y. A novel image security technique based on nucleic acid concepts. *J. Inf. Secur. Appl.* **2020**, *53*, 102505. [CrossRef]
36. Li, T.; Shi, J.; Li, X.; Wu, J.; Pan, F. Image encryption based on pixel-level diffusion with dynamic filtering and DNA-level permutation with 3D Latin cubes. *Entropy* **2019**, *21*, 319. [CrossRef] [PubMed]
37. Mezatio, B.A.; Motchongom Tingue, M.; Kengne, R.; Tchagna Kouanou, A.; Fozin Fonzin, T.; Tchitnga, R. Complex dynamics from a novel memristive 6D hyperchaotic autonomous system. *Int. J. Dyn. Control.* **2020**, *8*, 70–90. [CrossRef]
38. Li, T.; Zhou, M. ECG Classification Using Wavelet Packet Entropy and Random Forests. *Entropy* **2016**, *18*, 285. [CrossRef]
39. Li, T.; Qian, Z.; He, T. Short-term load forecasting with improved CEEMDAN and GWO-based multiple kernel ELM. *Complexity* **2020**, *2020*, 1209547. [CrossRef]
40. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Song, Y.; Xu, J. An improved differential evolution algorithm and its application in optimization problem. *Soft Comput.* **2021**, 1–22. [CrossRef]
41. Song, Y.; Wu, D.; Deng, W.; Gao, X.; Li, T.; Zhang, B.; Li, Y. MPPCEDE: multi-population parallel co-evolutionary differential evolution for parameter optimization. *Energy Conv. Manag.* **2021**, *228*, 113661. [CrossRef]
42. Li, T.; Zhou, M.; Guo, C.; Luo, M.; Wu, J.; Pan, F.; Tao, Q.; He, T. Forecasting crude oil price using EEMD and RVM with adaptive PSO-based kernels. *Energies* **2016**, *9*, 1014. [CrossRef]
43. Deng, W.; Xu, J.; Cai, X.; Song, Y.; Zhao, H. Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem. *Appl. Soft. Comput.* **2021**, *100*, 106724. [CrossRef]
44. Liu, L.; Zhang, L.; Jiang, D.; Guan, Y.; Zhang, Z. A Simultaneous Scrambling and Diffusion Color Image Encryption Algorithm Based on Hopfield Chaotic Neural Network. *IEEE Access* **2019**, *7*, 185796–185810. [CrossRef]
45. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137. [CrossRef]
46. Mohamed, H.G.; ElKamchouchi, D.H.; Moussa, K.H. A Novel Color Image Encryption Algorithm Based on Hyperchaotic Maps and Mitochondrial DNA Sequences. *Entropy* **2020**, *22*, 158. [CrossRef]
47. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [CrossRef]
48. Özkaynak, F. Brief review on application of nonlinear dynamics in image encryption. *Nonlinear Dyn.* **2018**, *92*, 305–313. [CrossRef]
49. Murillo-Escobar, M.A.; Meranza-Castillón, M.O.; López-Gutiérrez, R.M.; Cruz-Hernández, C. Suggested Integral Analysis for Chaos-Based Image Cryptosystems. *Entropy* **2019**, *21*, 815. [CrossRef] [PubMed]