mathematics

MDPI

Article

Color Image Encryption Algorithm Based on Cross-Spiral Transformation and Zone Diffusion

Xiaoqiang Zhang *, Mi Liu and Xiaochang Yang

School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China; ts20060159p31@cumt.edu.cn (M.L.); ts20060219p31@cumt.edu.cn (X.Y.)
* Correspondence: zhangxiaoqiang@cumt.edu.cn

Abstract: Due to their rich information, color images are frequently utilized in many different industries, but the network's security in handling their delivery of images must be taken into account. To improve the security and efficiency of color images, this paper proposed a color image encryption algorithm based on cross-spiral transformation and zone diffusion. The proposed algorithm is based on Chen's system and the piecewise linear chaotic map, and uses the chaotic sequences generated by them for related operations. Firstly, the R, G and B planes are extracted, and the spiral starting point of each plane is randomly selected by the chaotic sequence to implement the cross-spiral transformation. Secondly, the bit-level image matrix is constructed by the scrambled image matrix, and the bit-level chaotic matrix is constructed by the chaotic sequence. Finally, the three-dimensional matrix is divided into four zones by a dividing line, and partition diffusion is carried out to obtain the encrypted image. Simulation results and algorithm analyses indicate that the proposed algorithm has superior performance and can resist a wide range of attacks.

Keywords: image security; spiral transformation; color image; chaotic system

MSC: 15A48; 15A51

Citation: Zhang, X.; Liu, M.; Yang, X. Color Image Encryption Algorithm Based on Cross-Spiral Transformation and Zone Diffusion. *Mathematics* **2023**, *11*, 3228. https://doi.org/10.3390/math11143228

Academic Editor: Lingfeng Liu

Received: 26 June 2023
Revised: 12 July 2023
Accepted: 20 July 2023
Published: 22 July 2023

1. Introduction

With the rise of emerging technologies, people have realized the rapid information transmission in Internet. Digital images have a significant role in communication, the medical industry, the military, and other fields as vital carriers of multimedia communications. However, the convenience brought by new technologies is accompanied by the risk of information leakage, which poses a threat to the development of the country, society, and individuals. In March 2022, Samsung Electronics was attacked by a hacker group, resulting in the leakage of a large amount of the company's confidential data. That same month, the anonymous hacking group released a database of food giant Nestlé on its Twitter account, exfiltrating about 10 GB of sensitive data, including company emails, passwords, and data related to business customers. The consequences and losses caused by these information breaches are immeasurable. Therefore, the secure transmission of information is an urgent problem. As a result, both in theory and in practice, the protection of digital image information becomes crucial. Encrypting the plain image is the most common method.

Traditional encryption algorithms are mainly designed for protecting the security of text information, such as the advanced encryption standard [1], data encryption standard [2], and international data encryption algorithm [3]. However, due to the high redundancy and strong correlation of images, there are some drawbacks to encrypt the image by the traditional encryption algorithms, such as the low encryption efficiency and weak security [4,5]. With the wide application of digital images, these traditional encryption algorithms are obviously no longer applicable, so it is necessary to study the encryption algorithm suitable for digital images to ensure the secure and efficient transmission of images.

The encrypted image is obtained through the encryption algorithm with the encryption key. The encrypted image is transmitted to the recipient through the communication channel, and the key is transmitted through the secure channel. After the recipient finally obtains the encrypted image, the decryption algorithm and the decryption key are used to decrypt the encrypted image to obtain the plain images. During the image transmission, even if the encrypted image is attacked, it is difficult for the attacker to obtain the real information of the plain image without the decryption algorithm and decryption key. Most of the image encryption algorithms are designed based on the scrambling and diffusion operations, and the decryption algorithm is the inverse process of the encryption algorithm.

Most image encryption algorithms cover roughly two parts. The first part is the method design of generating chaotic sequences. The generation of chaotic sequences depends on chaotic systems and keys, and complex chaotic systems and excellent key-generated methods always make encryption algorithms more secure. The chaotic system is a nonlinear phenomenon with the characteristics of initial sensitivity, unpredictability, ergodicity, etc. [6]. It is very consistent with the concepts in cryptographic algorithms. Most of the current image encryption algorithms are based on chaotic theory [7–9]. Therefore, it is necessary to select a chaotic system with excellent performance and design a reasonable random sequence generation method. Fridrich firstly used chaotic theory in 1998 to change the positions of image pixels to achieve the purpose of encrypting images [10]. Wang et al. used the improved one-dimensional (1D) Logistic map to scramble the pixel position [11]. Naskar P. K. et al. used the Logistic map for diffusion [12]. Many scholars propose a 1D chaotic system. In addition, some experts and scholars use high-dimensional chaotic systems to encrypt images. Chen applied the Lorentz chaotic system to image encryption [13]. Luo et al. proposed a hybrid system [14].

The second part is the structural design of the encryption algorithm. The scrambling-diffusion mechanism is the framework of typical image encryption. The scrambling operation can change the pixel positions of the plain image to reduce the correlation of adjacent pixels. The classical scrambling methods include the spiral transformation [15], Zigzag transformation [16], Arnold transformation [17], magic square transformation [18], and Latin square transformation, etc. [19]. Among them, the magic square transformation and the Latin square transformation are complex. Arnold has a short conversion period and low efficiency. The Zigzag transformation has the disadvantages of the unchanged position of the first and last elements before and after the transformation and a single scanning starting point. Compared with the above transformations, the spiral transformation has the advantages of simple transformation and low time complexity, so the spiral transformation is selected to realize the chaotic process in our algorithm. Xian et al. proposed a novel chaotic image encryption algorithm based on the spiral transformation [15]. Tang et al. designed a double helix transformation that effectively shuffles the pixels of image blocks [20]. Yuan et al. devised a bit-level spiral-filling method that scans pixels with odd and even lines successively [21]. Wang et al. constructed a two-way spiral transformation. For the R, G, and B planes of color images, the left half of the region is scanned clockwise spirally, and then the right half of the area is counterclockwise. The scrambling effect of the proposed bidirectional spiral transformation is significantly better than that of the traditional spiral transformation [22]. Thangaraja et al. designed a randomly selectable starting point of the spiral based on the chaotic map and used this helical transformation to scramble the plain image in a clockwise direction [23]. Xiao et al. screwed the image in pixel blocks, and its starting position, orientation and direction were all controlled by the chaotic sequences [24]. Huang et al. firstly performed a clockwise or counterclockwise spiral transformation within the block on the image and then performed the spiral transformation between the blocks to achieve scrambling of the image [7]. Wang et al. designed a dynamic spiral scrambling algorithm to dynamically combine the chaotic sequence with the plain image to change the pixel value of the plain image. Experimental simulation analysis showed that the algorithm can resist various common attacks. However the non-square image needs to be filled, and there may be blank information after decryption [25]. Liu et al.

proposed the RSA algorithm to protect the structural parameters and geometric size of the structured spiral phase mask and the security of the JTC cryptosystem, which can be enhanced simultaneously [26]. Xian et al. proposed a novel chaotic image encryption algorithm with a spiral transformation-based fractal sorting matrix [27]. Xu et al. proposed a robust image encryption algorithm combining a new chaotic system and discrete cosine transformation. The spatial image is scrambled by the spiral transformation, and then the diffusion operation is performed to obtain the encrypted image [28]. Wang et al. designed a dynamic spiral block scrambling to encrypt the sparse matrix generated by performing discrete wavelet transformation (DWT) on the plain image. Then, the encrypted image is compressed and quantified to obtain the noise-like cipher image [29].

The diffusion operation can modify the pixel values of an image to improve the ability of statistical attacks. The scrambled image is still difficult to resist statistical attacks, and the security is not strong enough. Therefore, the further diffusion operation on the image is necessary. Diffusion methods processes generally include the exclusive OR (XOR) operation diffusion and additive mode diffusion. To further strengthen the diffusion method, Huang et al. carried out the XOR diffusion of the three components of the color image by a cyclic shift of the row and column, respectively [30]. Zhu et al. constructed an improved two-dimensional (2D) diffusion structure that extends slight variations of the plain image to the entire encrypted image [31]. The diffusion process can also be divided into pixel-level diffusion and bit-level diffusion from the study of particle size. Bit-level image encryption causes both the positions and pixel values in the image matrix to change. Wang et al. devised a snake-like pixel-level diffusion method. The chaotic image is XOR from left to right, the row elements move in a circular manner, and then the columns of the image are XOR from right to left in a serpentine order, and the column elements move in a circular manner [32]. Xu et al. designed a bit-level mutual diffusion technique, which can achieve the ideal effect in just one round [33]. Wang et al. designed cross-plane diffusion rule based on the bit-level level and combined it with the S-box to replace half a pixel. The algorithm design is ingenious, and the secure factor is high [34].

At present, many encryption algorithms for color images have been proposed in academia. Wang et al. innovated a chaotic system and applied it to color images in combination with new deterministic scrambling and XOR diffusion [35]. Zhang et al. combined the three channels of color images into a 2D matrix. They used a chaotic index to scramble the pixel positions and use the DNA dynamic coding operation to obtain color-encrypted images [36]. Liu et al. used Arnold transformation to shuffle the pixels of R, G, and B components, and then they spread the pixel values with the help of chaotic sequences [37].

However, the current color image encryption algorithm still has shortcomings. For examples, the partial scrambling algorithm has an obvious horizontal correlation [38], and many algorithms only repeat the grayscale image encryption algorithm in the three components, R, G, and B, when encrypting color images, ignoring the high correlation among the R, G, and B components [36]. Therefore, on the basis of breaking the strong correlation between the components of color images and making full use of the characteristics of color images, it is necessary to propose an effective color image encryption algorithm for solving the problem of weak security and low efficiency. An image encryption algorithm based on cross-spiral transformation and zone segmentation is proposed in this paper. Firstly, to increase the key space and improve the key sensitivity, the hash value of the plaintext color image and six external parameters are utilized as the key in the key generation stage. Secondly, a cross-spiral transformation with arbitrary starting points is established by combining the features of color images, and the cross-spiral transformation of the three components is performed by the constructed index matrix. Thirdly, the scrambled image is decomposed into 8-bit planes, which are redivided into four zones. The bitwise XOR operations are performed in different directions for each zone to obtain the cipher image. Finally, in the simulation and performance testing stage, several metrics demonstrate

that the proposed algorithm is highly resistant to brute-force, statistical, robustness, and chosen-plaintext attacks.

The main contributions of this paper are described as follows, (1) Color image encryption algorithm based on cross-spiral transformation and zone diffusion is proposed. (2) The cross-spiral transformation is designed, which is not only suitable for any spiral starting point in the matrix but also breaks the strong correlation between the components of the color image. (3) The zone segmentation is designed, which can spread to three planes at the same time. The operation based on the bit level can make the desirable diffusion effect better.

The rest of this paper is arranged as follows. Section 2 describes the theoretical principles. Section 3 proposes a new color image encryption algorithm based on cross-spiral transformation and zone segmentation. Section 4 carries out some experiments. The experimental analyses are provided in Section 5. Section 6 draws the conclusions and outlooks.

2. Theoretical Principles

2.1. Color Image Encryption Algorithm

Vivid color images are widely used in various fields. The color image can be broken down into three components: red, blue, and green. They are arranged in a certain order and can be regarded as a three-dimensional (3D) matrix, as shown in Figure 1. All three components are integers ranging from 0 to 255. each pixel contains information about these three colors, and their proportions determine the color of the pixel.

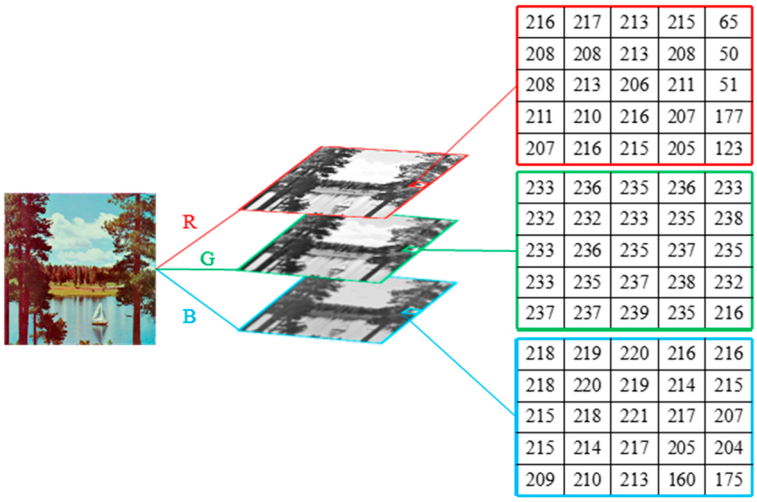


Figure 1. Color digital image.

There are two main ideas for existing color image encryption algorithms.

(1) The three channels are encrypted independently. Firstly, the three components, R, G, and B, of the color image are decomposed, as shown in Figure 2. Secondly, the same algorithm is used for the three channels and encrypted separately in the form of grayscale images. Finally, the grayscale ciphertext images of the three components are combined in their original order to form the final color ciphertext image. This line of thinking does not take into account the high correlation between the R, G, and B components, resulting in color images being slightly less defensive against attacks.

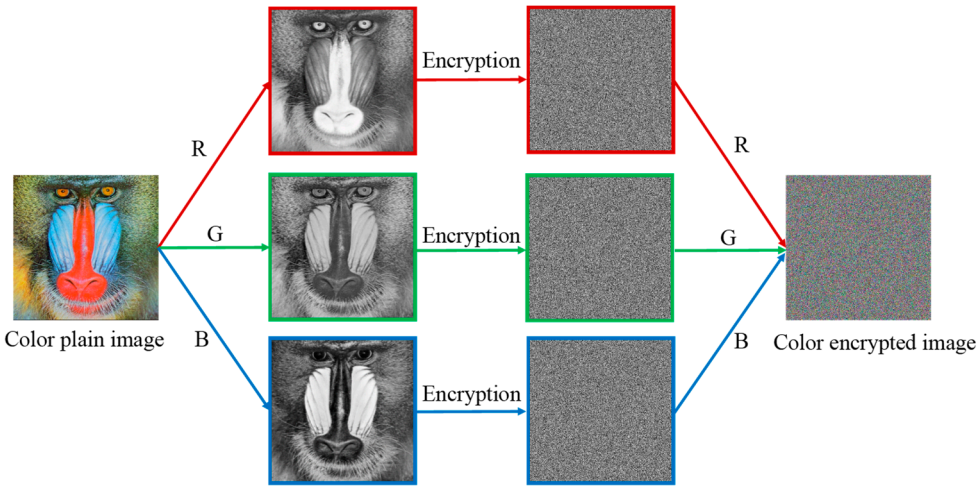


Figure 2. The three components are encrypted independently.

(2) The three channels are encrypted in the form of grayscale images as a whole. Firstly, the three channels, R, G, and B, of the color image are decomposed, as shown in Figure 3. Secondly, the three channels are first stitched into a large grayscale image. Finally, the whole is encrypted in the form of a grayscale image to obtain a ciphertext image. This line of thinking would ignore the characteristics of color images.

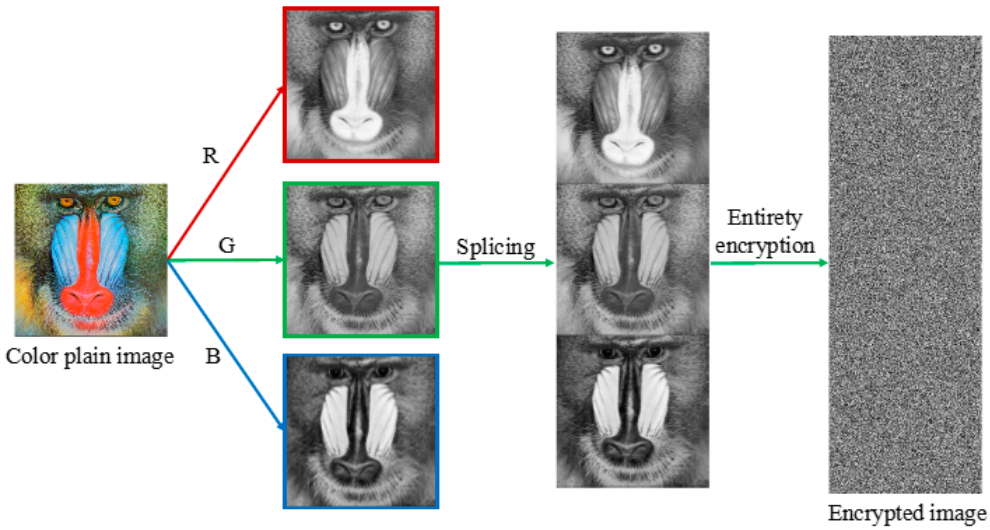


Figure 3. The three components are encrypted as a whole.

2.2. Traditional Spiral Transformation

The traditional spiral transformation is a classic method of transforming the position of pixels and can scan all pixels in the image in a spiral manner at a given spiral start and direction to complete the scrambling process [15]. As shown in Figure 4, the upper-right corner is used as the starting point to scramble the image. The matrix elements are changed following the arrow trajectory, which converts the spiral matrix into a vector of length 25. The two matrices of 5×5 shown are illustrated to represent the matrices before and after the spiral transformation, respectively.

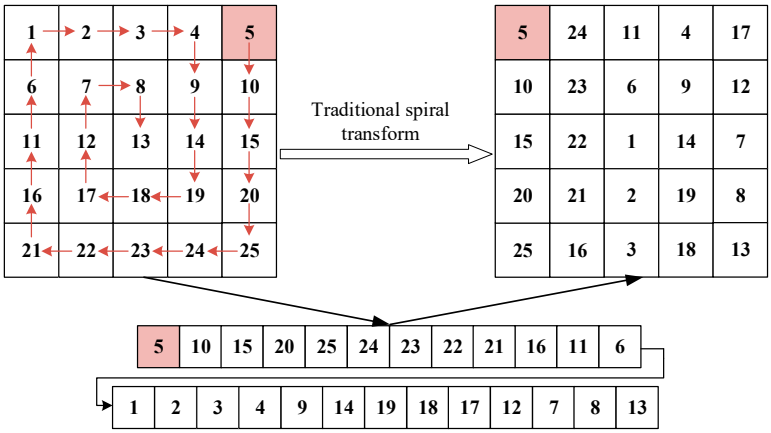


Figure 4. Traditional spiral transformation.

2.3. Cross Spiral Transformation

On the basis of the traditional spiral transformation, the, R, G, and B components and chaotic sequences of color images are combined to propose a new cross spiral transformation. The detailed steps are as described follows.

Step 1: Selecting the transformation starting points.

According to the chaotic sequence, the starting points are randomly selected in the three components: R, G, and B.

Step 2: Arbitrary points spiral transformation.

Although the traditional spiral transformation operation is efficient, it can only process square matrices, and the scrambling effect is undesirable. Therefore, an arbitrary points spiral transformation is designed to solve the problems of limited image size and weak scrambling effect. The specific transformation process is shown in Figure 5. Elements in the matrix are scanned clockwise from the selected scan starting point from the inside to the outside until the traversal is complete. The scanned elements are stored sequentially in a 2D matrix to obtain the scrambled matrix.

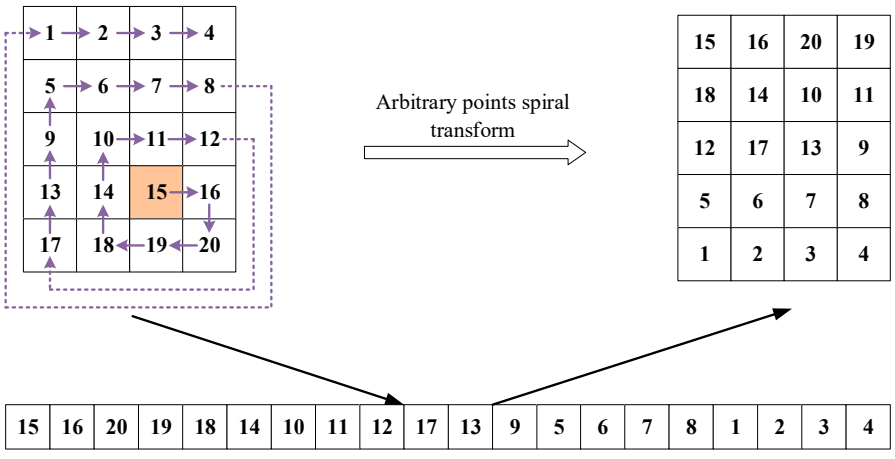


Figure 5. Arbitrary points spiral transformation.

In the R, G, and B components, the selected starting points are used to carry out a clockwise arbitrary points spiral transformation, and the scrambled R, G, and B components can be obtained.

Step 3: Cross-scrambling operation.

Three chaotic matrices, whose elements are 1, 2, or 3, are used to further scramble pixel position. The pixels in the scrambled R, G, and B components are selected, in turn, to obtain three new matrices and reconstitute a 3D matrix. If the element of the chaos matrix

is 1, it means that the element on the R component is selected. If the element of the chaos matrix is 2, it means that the element on the G component is selected. If the elements of the chaos matrix are 3, it means that the elements on the B component are selected.

Taking an example with a $5 \times 5 \times 3$ matrix, the specific transformation process of the cross-spiral transformation can be seen in Figure 6.

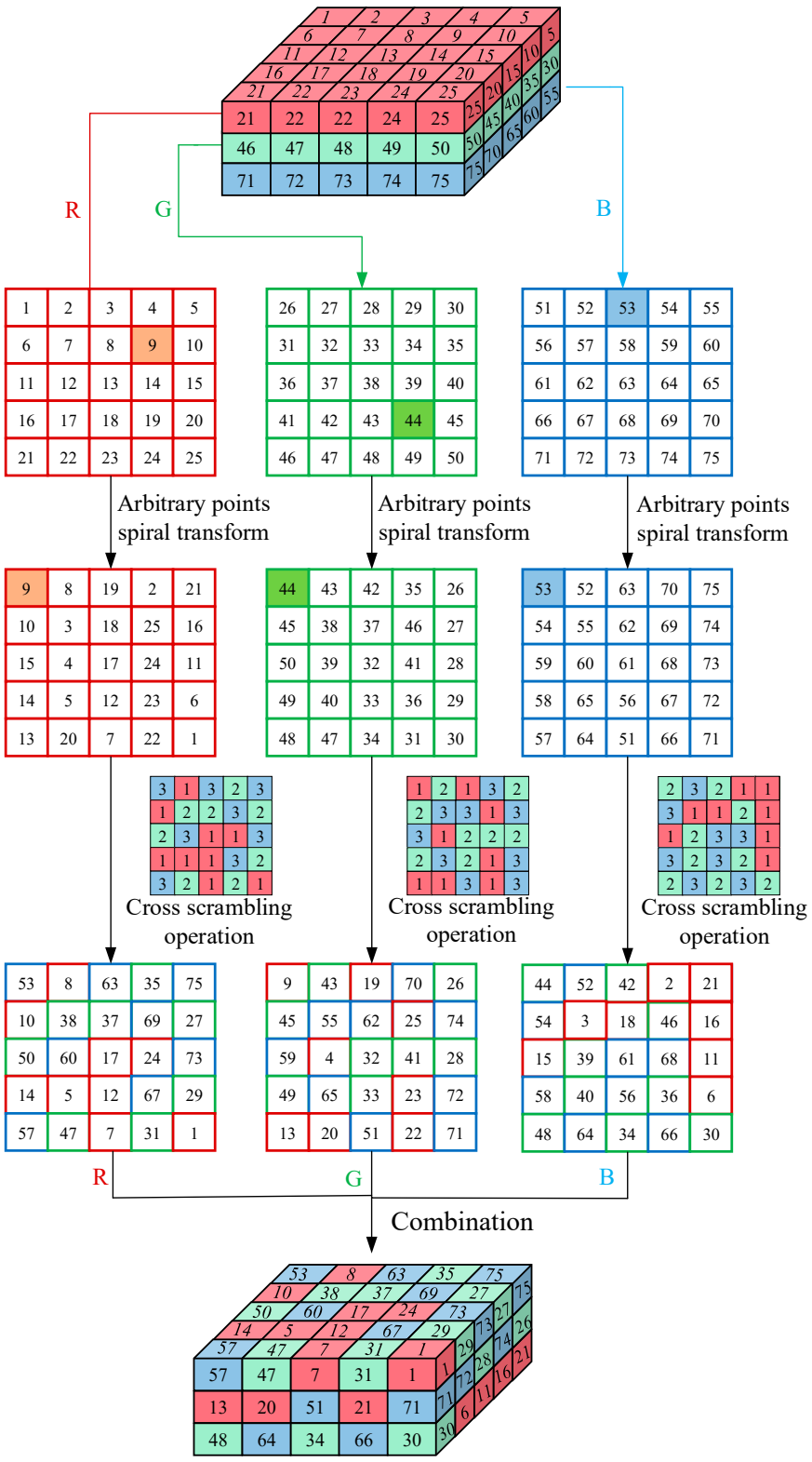


Figure 6. Cross-spiral transformation.

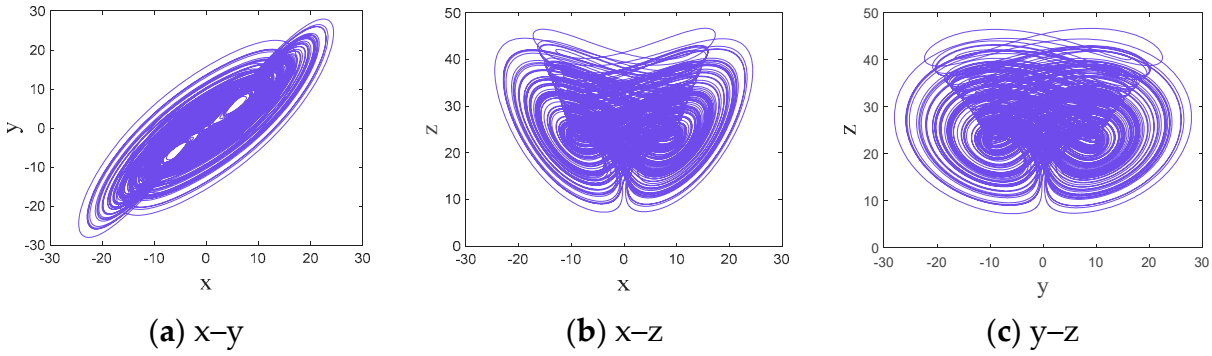
Firstly, arbitrary points spiral transformations are performed in the R, G, and B components in the $5 \times 5 \times 3$ matrix. Secondly, the chaotic matrix is used to realize cross-scrambling operations. Finally, the new R, G, and B components are combined to obtain a scrambled matrix.

2.4. Chen's Chaotic System

Chen's chaotic system has the complex dynamic behavior, and its chaotic sequences are random and unpredictable. Chen's chaotic system is defined by [39]:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = (c - a)x - xz + cy, \\ \dot{z} = xy - bz \end{cases} \tag{1}$$

where x, y, and z are state variables and a, b, and c are control parameters. This system behaves as a chaotic characteristic when $a = 35$, $b = 3$, and $20 \leq c \leq 28.4$ [39,40]. The chaotic attractor is shown in Figure 7. It can be seen that the chaotic system has excellent traversability when $c = 28$. Figure 8 shows a time series plot of Chen's map. Chen's map has uniform distribution and excellent traversability, which can provide a good random sequence for the encrypting of images.



(**a**) x–y (**b**) x–z (**c**) y–z

Figure 7. Attractors in Chen's chaotic system: (**a**) x–y; (**b**) x–z; (**c**) y–z.

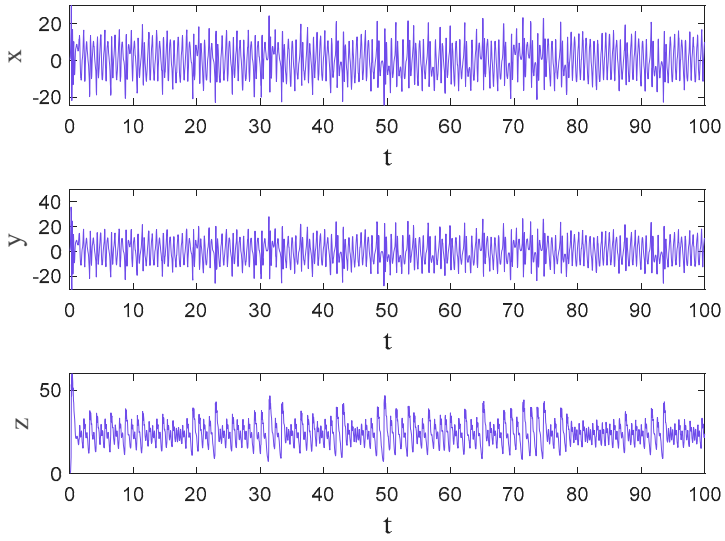


Figure 8. Time series distribution in Chen's chaotic system.

2.5. Piecewise Linear Chaotic Map

The piecewise linear chaotic map (PWLCM), as one of the common 1D chaotic systems, meets the characteristics of transversality and simplicity. The PWLCM is defined by [41]

$$s_{n+1} = f(s_n,\ p) = \begin{cases} s_n/p, & 0 < s_n < p \\ (s_n - p)/(0.5 - p), & p \le s_n < 0.5, \\ f(1 - s_n,\ p), & 0.5 \le s_n < 1 \end{cases} \tag{2}$$

where state variables are $s_n \in (0,\ 1)$ and control parameters are $p \in (0,\ 0.5)$. Figure 9 shows the bifurcation diagram of the PWLM.

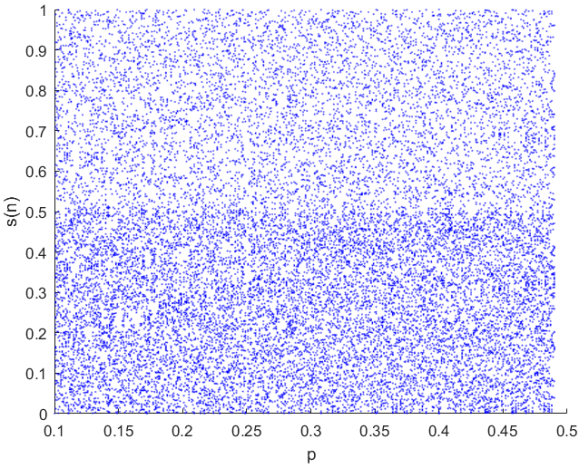


Figure 9. Bifurcation diagram of the PWLCM.

2.6. Zone Segmentation

The pixel values can be changed by the diffusion operation. If the bit-level diffusion method is reasonably designed, a better diffusion effect will be achieved to affect the entire image, and there will only be minor changes in the plain image. To improve the security of the proposed image encryption algorithm, this paper designs a bit-level zone segmentation method. Zone segmentation is an important part of the bit-level diffusion method. The main purpose is to divide the 3D image matrix into four zones, and each zone selects different diffusion methods to change the pixel values and enhance the diffusion effect.

A color image with a size of $m \times n \times 3$, and any pixel of the R, G, and B components, can be represented as having 8 bits. Therefore, a color image can be viewed as a 3D matrix, like Figure 10. Segmentation is performed by selecting a dividing line, which then divides the bit image matrix into four zones based on the chosen dividing line. The detailed steps for zone segmentation are described as follows.

Step 1: Bit-plane decomposition.

The plain color image is I with a size of $m \times n \times 3$. It is decomposed into $m \times n \times$ 24-bit planes. Therefore, I can be viewed as a 3D matrix T with a size of $m \times n \times 24$.

Step 2: Selecting the dividing line.

The dividing line $dpx \in \{1, 2, \ldots, m\}$ on the x-axis and the dividing line $dpy \in \{1, 2, \ldots, n\}$ on the y-axis are randomly selected by the chaotic sequence.

Step 3: 3D matrix segmentation.

According to the two dividing lines dpx and dpy, the 3D matrix T is decomposed into four zones: Z_1, Z_2, Z_3, and Z_4.

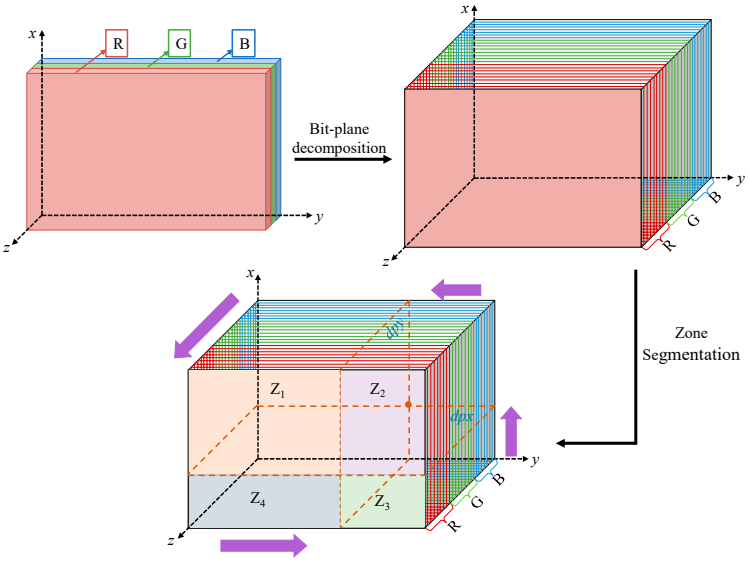


Figure 10. Schematic diagram of zone segmentation.

3. Algorithm Description

The proposed algorithm in this paper is composed of three stages, i.e., key generation, encryption process, and decryption process.

3.1. Key Generation

The SHA-256 of the plain color image and external parameters are used to generate the control parameters and initial values of Chen's chaotic system and the PWLCM. The detailed steps to generate the key are described as follows.

Step 1: Dividing the hash value.

The hash value K of the plain color image is decomposed into 32 segments with an 8-bit length:

$$K = k_1,\ k_2,\ \ldots,\ k_{32}. \tag{3}$$

Step 2: Calculating intermediate parameters.

The six intermediate parameters are generated by:

$$\begin{cases} h_1 = floor\left(\sum_{i=1}^{6}\delta_i + \frac{(k_1+k_3+k_5+k_7+k_9+k_{11})}{\max(k_1,k_3,k_5,k_7,k_9,k_{11})}\right) \\ h_2 = floor\left(\sum_{i=1}^{2}\delta_i + h_1 + \frac{(k_2+k_4+k_6+k_8+k_{10}+k_{12})}{\max(k_2,k_4,k_6,k_8,k_{10},k_{12})}\right) \\ h_3 = floor\left(\sum_{i=1}^{3}\delta_i + \sum_{i=1}^{2}h_i + \frac{(k_{13}+k_{15}+k_{17}+k_{19}+k_{21})}{\max(k_{13},k_{15},k_{17},k_{19},k_{21})}\right) \\ h_4 = floor\left(\sum_{i=1}^{4}\delta_i + \sum_{i=1}^{3}h_i + \frac{(k_{14}+k_{16}+k_{18}+k_{20}+k_{22})}{\max(k_{14},k_{16},k_{18},k_{20},k_{22})}\right) \\ h_5 = floor\left(\sum_{i=1}^{5}\delta_i + \sum_{i=1}^{4}h_i + \frac{(k_{23}+k_{25}+k_{27}+k_{29}+k_{31})}{\max(k_{23},k_{25},k_{27},k_{29},k_{31})}\right) \\ h_6 = floor\left(\sum_{i=1}^{6}\delta_i + \sum_{i=1}^{5}h_i + \frac{(k_{24}+k_{26}+k_{28}+k_{30}+k_{32})}{\max(k_{24},k_{26},k_{28},k_{30},k_{32})}\right) \end{cases}, \tag{4}$$

where $floor(\cdot)$ is the rounding toward negative infinity function, $\max(\cdot)$ means the maximum value in all numbers, and δ_1, δ_2, δ_3, δ_4, δ_5, and δ_6 are the external parameters.

Step 3: Generating initial values and control parameters.

Six intermediate parameters can be used to calculate the control parameter c and initial values x_0, y_0, and z_0 of Chen's chaotic system and the control parameter p and initial value s_0 of the PWLCM. These initial values and control parameters are generated by:

$$\begin{cases} x_0 = \left((h_1 \oplus h_2 \oplus h_3) / \sum_{i=1}^{6} h_i \right) \text{mod} 1 \\ y_0 = \left((h_2 \oplus h_3 \oplus h_4) / \sum_{i=1}^{6} h_i \right) \text{mod} 1 \\ z_0 = \left((h_3 \oplus h_4 \oplus h_5) / \sum_{i=1}^{6} h_i \right) \text{mod} 1 \\ s_0 = \left((h_4 \oplus h_5 \oplus h_6) / \sum_{i=1}^{6} h_i \right) \text{mod} 1 \\ p = (h_2 + h_4 + h_6) \text{mod} 0.4 + 0.1 \\ c = floor((h_1 + h_3 + h_5) \text{mod} 8.4) + 20 \end{cases}, \quad (5)$$

where $\oplus$ indicates XOR operation, and mod$(\cdot)$ denotes the modulus operation after division.

3.2. Encryption Process

The proposed algorithm uses the classical permutation-diffusion framework. In the scrambling stage, the chaotic sequence is used to select the spiral starting points of the three planes of R, G, and B in the color image, and then cross-spiral transformation is used to change the color image pixel position of each component to solve the problem of the strong correlation between pixels. In the diffusion stage, the chaotic sequence is used to randomly determine the two dividing lines, and then the bit-level zone segmentation is used to enhance the resistance to data statistical attacks. The encryption diagram is described in Figure 11. The specific steps are shown as follows.

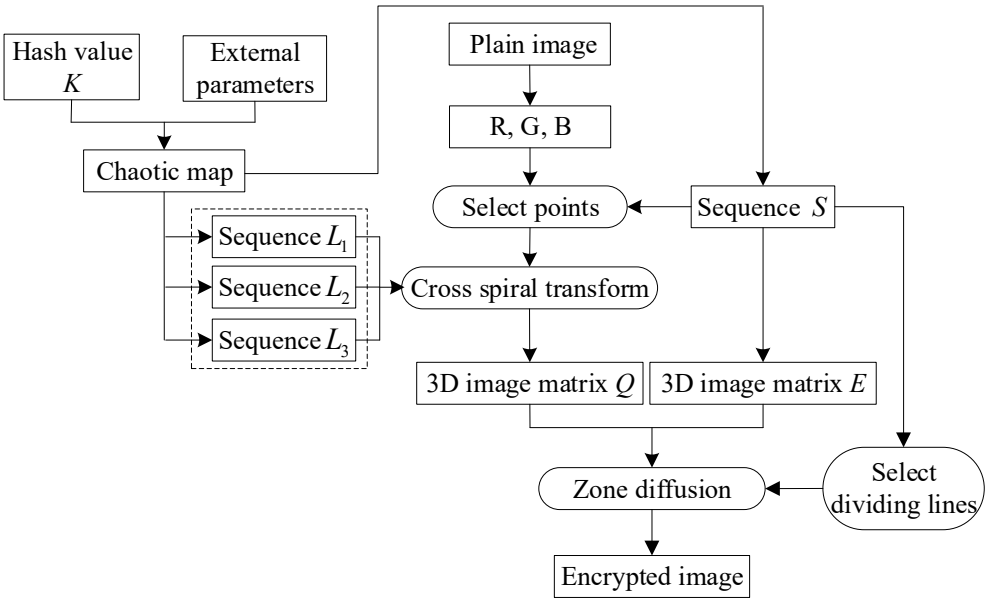


Figure 11. Block diagram of color image encryption process.

Step 1: Inputting the color image.

Let the plain color image be I, whose size is $m \times n \times 3$. Its R, G, and B components, IR, IG, and IB, are matrices with sizes of $m \times n$.

Step 2: Generating chaotic sequences.

This step uses the control parameter c and the initial values x_0, y_0, and z_0 of Chen's chaotic system defined in Section 2.4. Three chaotic sequences, L_1, L_2, and L_3, with lengths of mn, can be obtained after iterating 1000 + mn times according to Equation (1) and discarding the first 1000 values. Similarly, using the control parameter p and the initial

values s_0 of the PWLCM defined in Section 2.5, the chaotic sequence S with a length of $3\,mn + 8$ can be obtained after iterating $1008 + 3\,mn$ times according to Equation (2) and discarding the first 1000 values.

Step 3: Chaotic sequence processing.

L_1, L_2, and L_3 are processed by:

$$H_1(i) = \begin{cases} 1, & L_1(i) > L_2(i) \\ -1, & L_1(i) \le L_2(i) \end{cases}, \quad i = 1,\ 2,\ \cdots,\ mn, \tag{6}$$

$$H_2(i) = floor\Big(\Big(L_1(i) \times 10^{14}\Big) \bmod 2\Big) + 1, \quad i = 1,\ 2,\ \cdots,\ mn, \tag{7}$$

$$H_3(i) = floor\Big(\Big(L_2(i) \times 10^{14}\Big) \bmod 2\Big) + 1, \quad i = 1,\ 2,\ \cdots,\ mn, \tag{8}$$

where H_1, H_2, and H_3 are the intermediate sequences used to generate the index matrices.

Step 4: Generating index matrices.

L_3, H_1, H_2, and H_3 are processed by:

$$A_1(i) = floor\Big(\Big(L_3(i) \times 10^{14}\Big) \bmod 3\Big) + 1, \quad i = 1, 2, \cdots, mn, \tag{9}$$

$$A_2(i) = \begin{cases} H_1(i) + A_1(i), & A_1(i) = 1 \\ H_2(i) + A_1(i), & A_1(i) = 3 \\ H_3(i) + A_1(i), & A_1(i) = 2 \end{cases}, \quad i = 1,\ 2,\ \cdots,\ mn\ , \tag{10}$$

$$A_3(i) = \begin{cases} 1, & (A_1(i) = 2 \& A_2(i) = 3) \,||\, (A_1(i) = 3 \& A_2(i) = 2) \\ 2, & (A_1(i) = 1 \& A_2(i) = 3) \,||\, (A_1(i) = 3 \& A_2(i) = 1) \\ 3, & (A_1(i) = 1 \& A_2(i) = 2) \,||\, (A_1(i) = 2 \& A_2(i) = 1) \end{cases}, \quad i = 1,\ 2,\ \cdots,\ mn, \tag{11}$$

where A_1, A_2, and A_3 index sequences with a size of mn. A_1, A_2, and A_3 are reshaped into three new 2D chaotic matrices, X_1, X_2, and X_3, with lengths of $m \times n$ to select the pixels of R, G, and B. The specific selection rules are described as follows.

(1) If the value of the index matrix X_1, X_2, and X_3 is 1, it means that the pixel of the R component is selected;
(2) If the value of the index matrix X_1, X_2, and X_3 is 2, it means that the pixel of the G component is selected;
(3) If the value of the index matrix X_1, X_2, and X_3 is 3, it means that the pixel of the B component is selected.

Step 5: Cross-spiral transformation.

The starting point of the spiral transformation of the three planes of R, G, and B is calculated according to the first six values of the chaotic sequence S. The specific formula is designed by:

$$sp_i = floor\Big(\Big(S(i) \times 10^{14}\Big) \bmod 256\Big), \quad i = 1, 2, \cdots, 6, \tag{12}$$

where sp_i (i = 1, 2, . . ., 6) represents the coordinates of the spiral transformation starting in the three planes of R, G, and B. The starting point in the R plane is (sp_1, sp_2). The starting point in the G plane is (sp_3, sp_4). The starting point in the B plane is (sp_5, sp_6).

The cross-spiral transformation is achieved in *IR*, *IG*, and *IB* to obtain three scrambled matrices, P_1, P_2, and P_3 with a size of mn, using sp_i, X_1, X_2, and X_3.

Step 6: Zone segmentation.

Each pixel value of P_1, P_2, and P_3 is converted from a decimal to 8-bit binary to obtain three binary matrices, Q_1, Q_2, and Q_3, with sizes of $mn \times 8$. Q_1, Q_2, and Q_3 are reshaped into a 3D image matrix Q with a size of $m \times n \times 24$.

S is calculated to obtain X with a length of 3 mn, and X is converted into a binary 3D chaotic matrix B with a size of $m \times n \times 24$.

$$X(i) = floor\Big(\Big(S(i) \times 10^{14}\Big) \bmod 256\Big), \quad i = 9, 10, \cdots, 3mn + 9. \tag{13}$$

S is used to select the dividing lines $x = dpx$ on the x-axis and $y = dpy$ on the y axis; they are calculated by:

$$\begin{cases} dpx = \big(floor\big(S(7) \times 10^{14}\big)\big) \bmod m \\ dpy = \big(floor\big(S(8) \times 10^{14}\big)\big) \bmod n \end{cases}, \tag{14}$$

where $dpx \in (1, m)$, $dpy \in (1, n)$, dpx, and dpy are used as the two dividing lines to divide Q and E. They are divided into four areas: Z_1, Z_2, Z_3, and Z_4.

Step 7: Zone diffusion.

Row by row or column by column XOR operations are performed on each zone Q and E to obtain a 3D matrix W with a size of $m \times n \times 24$. The specific process of zone diffusion is designed by Z_1 zone diffusion, Z_2 zone diffusion, Z_3 zone diffusion, and Z_4 zone diffusion.

(1) Z_1 zone diffusion:

$$\begin{cases} W(i,j,z) = Q(i,j,z) \oplus Q(i+1,j,z) \oplus E(i,j,z), & i = dpx+1,\ dpx+2,\ \ldots,\ m-1 \\ W(i,j,z) = Q(i,j,z) \oplus E(i,j,z), & i = m \end{cases}, \tag{15}$$

(2) Z_2 zone diffusion:

$$\begin{cases} W(i,j,z) = Q(i,j,z) \oplus E(i,j,z), & j = dpy+1 \\ W(i,j,z) = Q(i,j,z) \oplus Q(i,j-1,z) \oplus E(i,j,z), & j = dpy+2,\ dpy+3,\ \ldots,\ n \end{cases}, \tag{16}$$

where $i = dpx + 1, dpx + 2, \ldots, m$, and $z = 1, 2, \ldots, 24$.

(3) Z_3 zone diffusion:

$$\begin{cases} W(i,j,z) = Q(i,j,z) \oplus Q(i,j,z+1) \oplus E(i,j,z), & z = 1,\ 2,\ \ldots,\ 23 \\ W(i,j,z) = Q(i,j,z) \oplus E(i,j,z), & z = 24 \end{cases}, \tag{17}$$

where $i = 1, 2, \ldots, dpx$, $y = dpy$ +1, and $dpy + 2, \ldots, n$.

(4) Z_4 zone diffusion:

$$\begin{cases} W(i,j,z) = Q(i,j,z) \oplus Q(i,j+1,z) \oplus E(i,j,z), & j = 1,\ 2,\ \ldots,\ dpy-1 \\ W(i,j,z) = Q(i,j,z) \oplus E(i,j,z), & j = dpy \end{cases}, \tag{18}$$

where $i = 1, 2, \ldots, dpx$ and $z = 1, 2, \ldots, 24$.

Step 8: Generating the encrypted image.

W is reshaped into a matrix with a size of 3 $mn \times 8$, and the matrix is converted to a decimal sequence U with a length of 3 mn. U reshaped the encrypted image C with a size of $m \times n \times 3$. Algorithm 1 shows the encryption process.

Algorithm 1: Encryption process.

Input: Plain color image I, h_1, h_2, h_3, h_4, h_5 and h_6
Output: Encryption image C

$x_0 = \left((h_1 \oplus h_2 \oplus h_3) / \sum_{i=1}^{6} h_i \right) \mathrm{mod} 1$

$y_0 = \left((h_2 \oplus h_3 \oplus h_4) / \sum_{i=1}^{6} h_i \right) \mathrm{mod} 1$

$z_0 = \left((h_3 \oplus h_4 \oplus h_5) / \sum_{i=1}^{6} h_i \right) \mathrm{mod} 1$

$s_0 = \left((h_4 \oplus h_5 \oplus h_6) / \sum_{i=1}^{6} h_i \right) \mathrm{mod} 1$

$p = (h_2 + h_4 + h_6) \mathrm{mod} 0.4 + 0.1$

$c = floor((h_1 + h_3 + h_5) \mathrm{mod} 8.4) + 20$

L_1, L_2, L_3= Chen (x_0, y_0, z_0, 1000 + 1: 1000 + $m \times n$)

for t = 1 to mn do:

$$H_1(i) = \begin{cases} 1, & L_1(i) > L_2(i) \\ -1, & L_1(i) \le L_2(i) \end{cases}$$

$H_2(i) = floor\left(\left(L_1(i) \times 10^{14} \right) \mathrm{mod} 2 \right) + 1$

$H_3(i) = floor\left(\left(L_2(i) \times 10^{14} \right) \mathrm{mod} 2 \right) + 1$

$A_1(i) = floor\left(\left(L_3(i) \times 10^{14} \right) \mathrm{mod} 3 \right) + 1$

$$A_2(i) = \begin{cases} H_1(i) + A_1(i), & A_1(i) = 1 \\ H_2(i) + A_1(i), & A_1(i) = 3 \\ H_3(i) + A_1(i), & A_1(i) = 2 \end{cases}$$

$$A_3(i) = \begin{cases} 1, & (A_1(i) = 2 \& A_2(i) = 3) \| (A_1(i) = 3 \& A_2(i) = 2) \\ 2, & (A_1(i) = 1 \& A_2(i) = 3) \| (A_1(i) = 3 \& A_2(i) = 1) \\ 3, & (A_1(i) = 1 \& A_2(i) = 2) \| (A_1(i) = 2 \& A_2(i) = 1) \end{cases}$$

end for

X_1 = *reshape* (A_1, m, n)

X_2 = *reshape* (A_2, m, n)

X_3 = *reshape* (A_3, m, n)

when X_1 = 1, X_2 = 1, X_3 = 1, R component is selected

when X_1 = 2, X_2 = 2, X_3 = 2, G component is selected

when X_1 = 3, X_2 = 3, X_3 = 3, B component is selected

for t = 1 to 3 do:

Pt= Cross spiral transform (I(:, :, t))

Qt = *dec*2*bin*(Pt)

C(:, :, t) = Zone diffusion Qt

Et = C(:, :, t)

end for

3.3. Decryption Process

Each part of the proposed algorithm is reversible, and the encrypted image can be decrypted to obtain the correct plain image by the correct decryption key and the reverse operations of the encrypted process. The decryption diagram is shown in Figure 12.

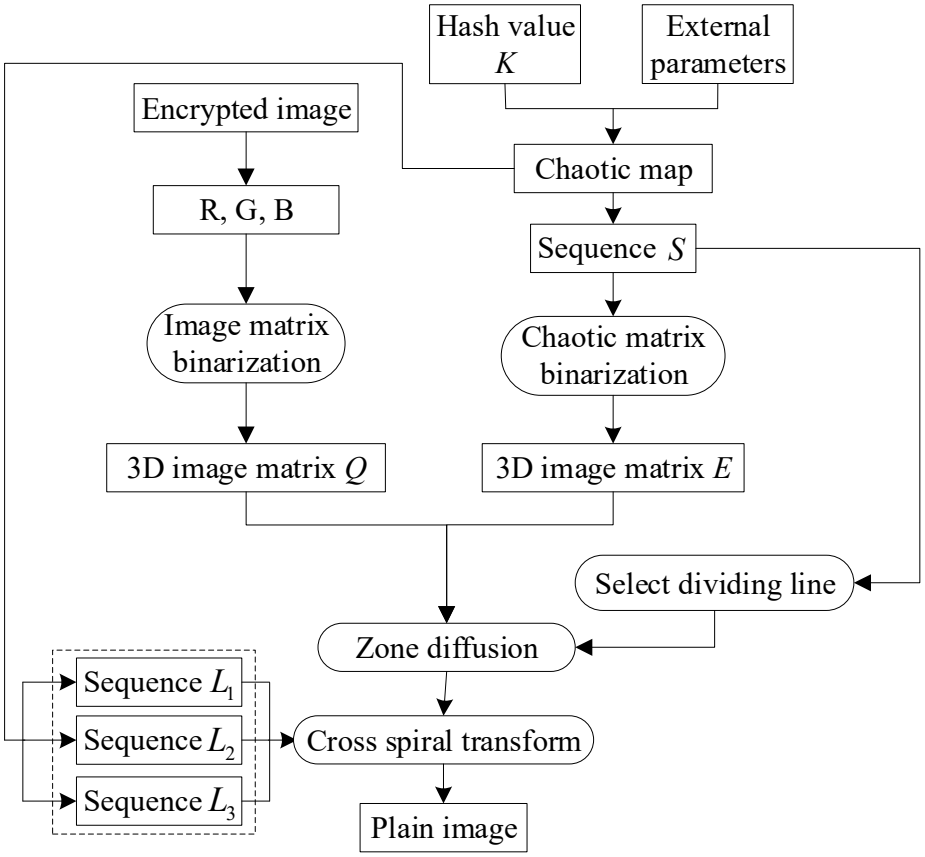


Figure 12. Block diagram of the color image decryption process.

4. Simulation Experiments and Results

The encryption algorithm is run in Matlab R2018b. The hardware environment is a 2.8 GHz CPU processor and 8 GB of memory. The software environment is a 64-bit Windows 10 operating system. The keys are hash value K of the plain color image and the external parameters $\delta_1 = 20$, $\delta_2 = 0.4$, $\delta_3 = 12$, $\delta_4 = 130$, $\delta_5 = 256$, and $\delta_6 = 1.6$. The plain images are converted into encrypted images through the proposed algorithm. We tested 50 images, 16 of which are shown in Figure 13. They are from the University of Southern California SIPI image database (http://sipi.usc.edu/database (accessed on 5 April 2023)) [42]. Figure 13a–p shows that sixteen color images from the database: Tree, Jelly beans, Couple, and Female are 256 × 256 × 3. Baboon, House, Peppers, and Airplane are 512 × 512 × 3. Sailboat has a size of 489 × 281 × 3 and Splash has a size of 377 × 467 × 3. Richmond, Foster City, Oakland, San Diego, Shreveport, and Stockton are 1024 × 1024 × 3. In Figure 14a–p, the encrypted images are displayed. The encryption images can be effectively restored using the decryption procedure, as shown in Figure 15a–p.

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

Figure 13. *Cont.*

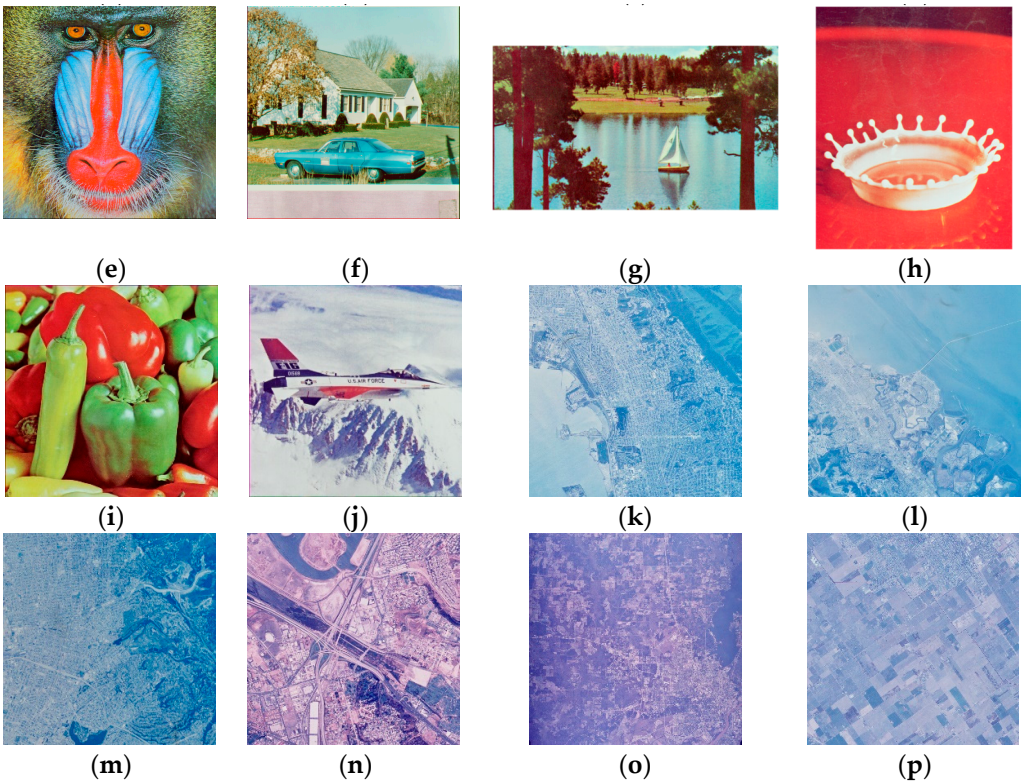


Figure 13. (**a**–**p**) Plain images: (a) Tree; (**b**) Jelly beans; (**c**) Couple; (**d**) Female; (**e**) Baboon; (**f**) House; (**g**) Sailboat; (**h**) Splash; (**i**) Peppers; (**j**) Airplane; (**k**) Richmond; (**l**) Foster City; (**m**) Oakland; (**n**) San Diego; (**o**) Shreveport; (**p**) Stockton.

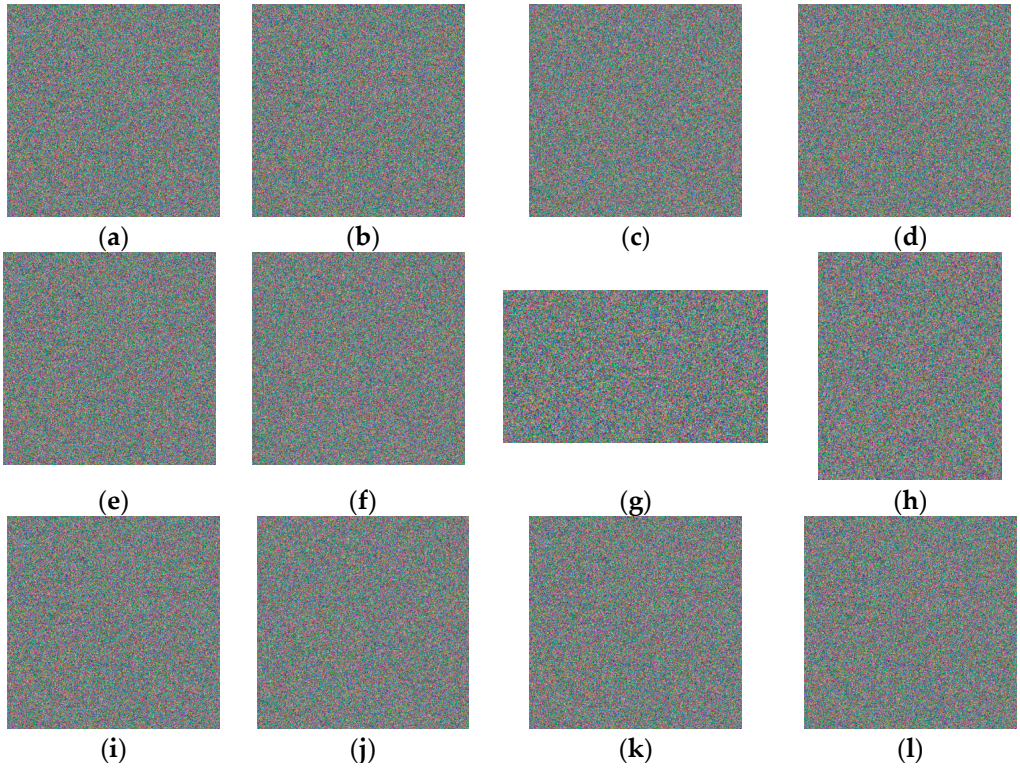
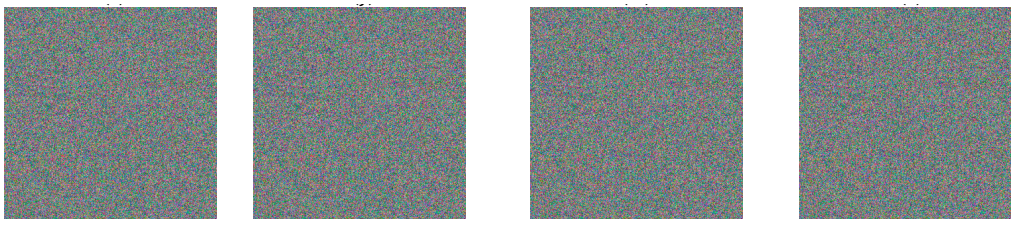


Figure 14. *Cont.*



(**m**) (**n**) (**o**) (**p**)

Figure 14. (**a**–**p**) Encrypted images: (a) Tree; (**b**) Jelly beans; (**c**) Couple; (**d**) Female; (**e**) Baboon; (**f**) House; (**g**) Sailboat; (**h**) Splash; (**i**) Peppers; (**j**) Airplane; (**k**) Richmond; (**l**) Foster City; (**m**) Oakland; (**n**) San Diego; (**o**) Shreveport; (**p**) Stockton.

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

(**e**) (**f**) (**g**) (**h**)

(**i**) (**j**) (**k**) (**l**)

(**m**) (**n**) (**o**) (**p**)

Figure 15. (**a**–**p**) Decrypted images: (a) Tree; (**b**) Jelly beans; (**c**) Couple; (**d**) Female; (**e**) Baboon; (**f**) House; (**g**) Sailboat; (**h**) Splash; (**i**) Peppers; (**j**) Airplane; (**k**) Richmond; (**l**) Foster City; (**m**) Oakland; (**n**) San Diego; (**o**) Shreveport; (**p**) Stockton.

5. Algorithm Analyses

5.1. Key Space Analysis

It is known that the key space is one of the important indicators to measure the security of the algorithms. The larger the key space, the stronger the resistance to brute force attacks. When the key size made by the algorithm exceeds 2^{100}, it can be considered to have the conditions to resist brute force attacks, and the algorithm is secure [43]. The keys are hash value K of the plain color image and the external parameters $\delta_1 = 20$, $\delta_2 = 0.4$, $\delta_3 = 12$,

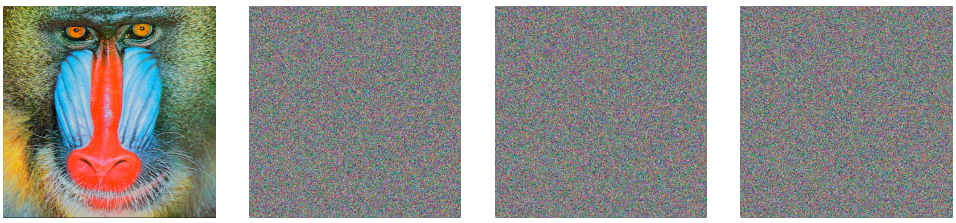
$\delta_4 = 130$, $\delta_5 = 256$, and $\delta_6 = 1.6$. The key space is $10^{14 \times 6} \times 2^{256} \approx 10^{161}$, and it is much larger than the minimum key space required. The comparisons of the proposed algorithm with the key space of other algorithms is shown in Table 1. The results show that compared with the existing algorithms, the proposed algorithm has better resistance to brute-force attacks.

Table 1. Key space analysis.

Algorithm	Proposed	Ref. [44]	Ref. [45]	Ref. [46]	Ref. [47]
Key space	10^{161}	10^{135}	10^{56}	10^{128}	10^{90}

5.2. Key Sensitivity Analysis

Analyzing the key sensitivity is a crucial indicator for confirming the algorithm. Key sensitivity refers to the impact of small changes in keys in the same encryption algorithm on producing results [48]. The stronger the key sensitivity, the greater the difference between the obtained result and the result obtained by the original key under a slight change in the key, and the higher the security of the algorithm. If the encrypted image is decrypted with two different keys. The decrypted results should be completely different. Figure 16a draws the image decrypted of a Baboon with the correct key. Figure 16b–d depicts the encrypted image of the Baboon using the incorrect key. These decrypted images have no visible connection to one another.



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

Figure 16. Key sensitivity test results: (**a**) decryption result with the correct key; (**b**–**d**) decryption result with the incorrect key.

The distinction between the two results is displayed in Table 2. The suggested keys are, consequently, sensitive.

Table 2. Difference between decryption results from slightly modified keys.

Figure	Decrypted Key	Pixel Difference Ratios
Figure 13a	the correct key	0.0%
Figure 13b	$\delta_1 + 10^{-14}$	99.7421%
Figure 13c	$\delta_3 + 10^{-14}$	99.6357%
Figure 13d	$\delta_5 + 10^{-14}$	99.2297%

5.3. Information Entropy Analysis

Information entropy is a key factor in measuring cryptographic algorithms. The information entropy increases with better random performance [49]. Its mathematical equation is:

$$H(I) = \sum_{i=0}^{255} p(m_i) \log_2 \frac{1}{p(m_i)}, \tag{19}$$

where $p(m_i)$ denotes the pixel gray level m_i.

Therefore, encryption algorithms should be designed with the entropy value as high as possible. Table 3 draws the entropy values of plain images and corresponding encrypted images. The data show that the entropy values of the encrypted images of the proposed algorithm are very close to the expected value of 8 and have an advantage over other

algorithms, which means that the statistical information of the plain images is successfully hidden. It can be confirmed that the proposed algorithm has a strong anti-entropy attack ability.

Table 3. Information entropy values.

Algorithm	Images	Entropy of Plain Images			Entropy of Encrypted Images		
		R	G	B	R	G	B
Proposed	Baboon	7.7066	7.4752	7.7522	7.9993	7.9991	7.9993
	House	7.4156	7.2294	7.4353	7.9993	7.9992	7.9993
	Sailboat	7.1927	7.5641	7.3057	7.9741	7.9746	7.9745
	Splash	6.3093	6.9206	5.9263	7.9990	7.9990	7.9988
	Average	7.1560	7.2973	7.1048	7.9929	7.9929	7.9929
Ref. [22]	Baboon	7.7066	7.4752	7.7522	7.9970	7.9974	7.9975
Ref. [44]	Baboon	7.7066	7.4752	7.7522	7.9970	7.9973	7.9973
Ref. [45]	Baboon	7.7066	7.4752	7.7522	7.9992	7.9994	7.9992
Ref. [46]	Baboon	7.7066	7.4752	7.7522	7.9972	7.9973	7.9974

In addition, local information entropy was introduced into the experiment to assess the randomness of local images. Table 4 shows the local information entropy of the color-encrypted images of the proposed algorithm. It can be seen that the local information entropy of the encrypted images is in the ideal value (7.9019, 7.9030), which indicates that the proposed algorithm cannot only encrypt the color image, but also have good randomness in the local image.

Table 4. Test results of local information entropy of the color-encrypted images.

Color Encrypted Images	Components	Local Information Entropy		Pass/File
		Test Values	Average Values	
Baboon	R	7.9020	7.9021	Passed
	G	7.9023		Passed
	B	7.9021		Passed
House	R	7.9026	7.8026	Passed
	G	7.9028		Passed
	B	7.9024		Passed
Sailboat	R	7.9026	7.9028	Passed
	G	7.9030		Passed
	B	7.9028		Passed
Splash	R	7.9025	7.9022	Passed
	G	7.9021		Passed
	B	7.9022		Passed

5.4. Histogram Analysis

The histogram, also known as the mass distribution map, represents the frequency of pixel values when analyzing the image and reflects the distribution of pixel values in the image [50]. Ideally, the histograms of the original image are not evenly distributed, and the pixel values of encrypted images occur almost identically; that is, the histograms are evenly distributed. Figure 17 shows that the histograms of the encrypted image are relatively evenly distributed. Therefore, the proposed algorithm can disrupt the image's pixel distribution.

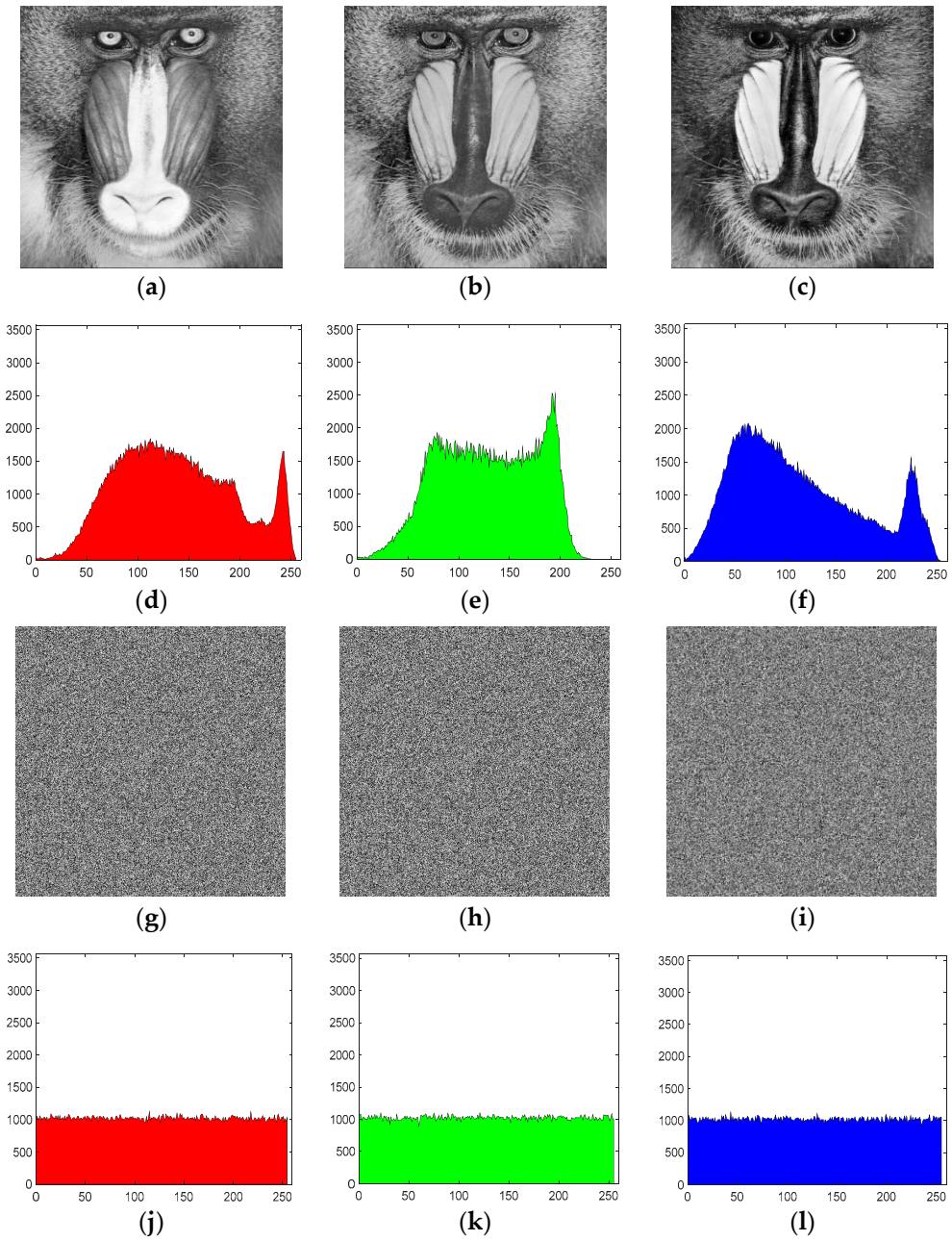


Figure 17. Histogram analysis of the Baboon image: (**a**) Baboon—R; (**b**) Baboon—G; (**c**) Baboon—B; (**d**) Histogram of (**a**); (**e**) Histogram of (**b**); (**f**) Histogram of (**c**); (**g**) Encrypted image—R; (**h**) Encrypted image—G; (**i**) Encrypted image—B; (**j**) Histogram of (**g**); (**k**) Histogram of (**h**); (**l**) Histogram of (**b**).

5.5. Differential Attack Analysis

Differential attack analysis tests the plaintext sensitivity of the algorithm aim to encrypt two original images with the same key that are only slightly different [51]. Two encrypted images can be obtained. A great algorithm should result in the ciphertext images having larger changes than before. The Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) can test the capability of a differential attack. The ideal values of the NPCR and UACI are about 99.61% and 33.46%, respectively [52].

Both the specific values can be obtained by [53]:

$$NPCR = \frac{1}{m \times n} \times \sum_{i=1}^{m} \sum_{j=1}^{m} D(i,j) \times 100\%, \tag{20}$$

$$UACI = \frac{1}{255 \times m \times n} \times \sum_{i=1}^{m} \sum_{j=1}^{n} \left| C(i,j) - C'(i,j) \right| \times 100\%, \tag{21}$$

where $C\,(i, j)$ is the unmodified plain image, and $C'(i, j)$ and $C\,(i, j)$ only have a one-pixel difference.

$$D(i,j) = \begin{cases} 0\ C(i,j) = C'(i,j) \\ 1\ C(i,j) \neq C'(i,j) \end{cases}. \tag{22}$$

The results are close to the ideal values, as shown in Table 5, and they show the proposed algorithm defends against differential attacks.

Table 5. The NPCR and UACI values of the color-encrypted images.

Algorithms	Images	NPCR (%)			UACI (%)		
		R	**G**	**B**	**R**	**G**	**B**
Proposed	Baboon	99.61	99.61	99.60	33.42	33.41	33.43
	House	99.59	99.61	99.61	33.41	33.42	33.44
	Sailboat	99.64	99.60	99.60	33.47	33.51	33.50
	Splash	99.62	99.62	99.60	33.50	33.64	33.51
Ref. [22]	Baboon	99.62	99.62	99.63	33.57	33.37	33.63
Ref. [44]	Baboon	99.61	99.55	99.60	33.45	33.41	33.28
Ref. [45]	Sailboat	99.59	99.62	99.60	33.47	33.46	33.48
Ref. [46]	Baboon	99.65	99.64	99.62	33.26	33.64	33.33

5.6. Correlation of Adjacent Pixels

Image transformation has a high degree of data redundancy. To prevent an attacker from analyzing the correlation of the adjacent pixels, the cipher image pixels should be as uncorrelated as possible [54]. That is, the coefficient should be close to the ideal value of 0. It is defined by:

$$D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2, \tag{23}$$

$$r_{x,y} = \frac{E((x - E(x))(y - E(y)))}{\sqrt{D(x)D(y)}}, \tag{24}$$

$$E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{25}$$

where x and y are the values of the adjacent pixels and $D(x)$ and $E(x)$ are the variance and mathematical expectation of x, respectively. The calculated results are revealed in Tables 6 and 7. Figures 18a–c, 19a–c and 20a–c show correlation maps of the plain color image Baboon and encrypted images of R, G, and B in three directions, respectively. It can be shown that the pixel correlation of the images before encryption is very high, and the pixel encrypted is uniform, and has a very low correlation. Figures 18d–f, 19d–f and 20d–f show that the plain images have a great large correlation, and the encrypted image of the proposed algorithm has a low correlation.

Table 6. Correlation coefficients of the plain images and corresponding encrypted images.

Images	Channel	Plain Images			Encrypted Images		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Baboon	R	0.9227	0.8597	0.8476	0.0004	−0.0003	0.00272
	G	0.8656	0.7578	0.7260	0.0038	0.0009	0.0042
	B	0.9070	0.8776	0.8357	−0.0010	0.0001	−0.0013
House	R	0.9543	0.9532	0.9184	−0.0025	−0.0013	−0.0006
	G	0.9339	0.9279	0.8771	0.0028	−0.0015	0.0019
	B	0.9751	0.9591	0.9356	0.0021	0.001	−0.0010
Sailboat	R	0.9415	0.9365	0.9203	0.0100	0.0359	0.0407
	G	0.9678	0.9664	0.9523	0.0162	0.0484	0.0510
	B	0.9691	0.9702	0.9511	0.0485	0.0810	0.1786
Splash	R	0.9883	0.9942	0.9862	0.0031	−0.0027	−0.0023
	G	0.9883	0.9877	0.9804	−0.0052	−0.0008	0.0019
	B	0.9864	0.9842	0.9753	0.0037	0.0021	−0.0008

Table 7. Comparisons of the correlation coefficients with other algorithms.

Images	Algorithms	Channel	Directions		
			Horizontal	Vertical	Diagonal
Encrypted images of the Baboon	Proposed	R	0.0004	−0.0003	0.0027
		G	0.0038	0.0009	0.0042
		B	−0.0010	0.0001	−0.0013
	Ref. [22]	R	−0.0017	−0.0007	0.0015
		G	0.0028	0.0039	0.0015
		B	0.0041	0.0061	0.0025
	Ref. [44]	R	0.0033	−0.0013	−0.0009
		G	0.0001	0.0020	−0.0012
		B	0.0000	0.0000	0.0004
	Ref. [45]	R	−0.0023	0.0014	0.0155
		G	−0.0115	−0.0178	0.0044
		B	0.0066	−0.0089	−0.0132
	Ref. [46]	R	−0.0036	−0.0109	−0.0052
		G	−0.0008	0.0070	0.0095
		B	−0.0009	0.0082	−0.0113

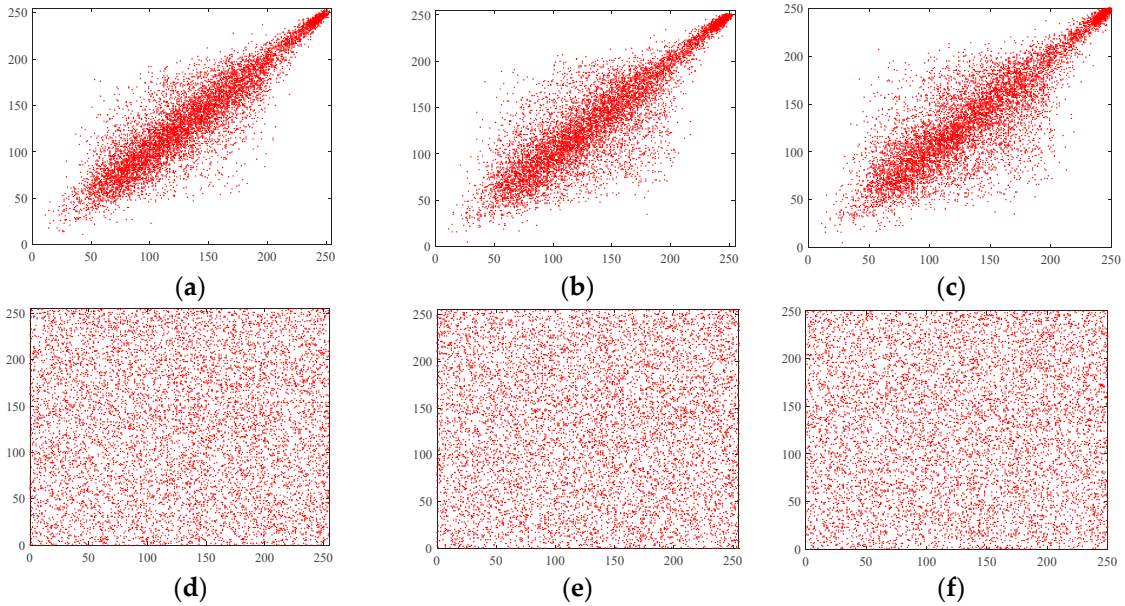


Figure 18. Correlation of the adjacent pixels on the R component of Baboon color image: (**a**) Baboon—R—Horizontal; (**b**) Baboon—R—Vertical; (**c**) Baboon—R—Diagonal; (**d**) Encrypted—R—Horizontal; (**e**) Encrypted—R—Vertical; (**f**) Encrypted—R—Diagonal.

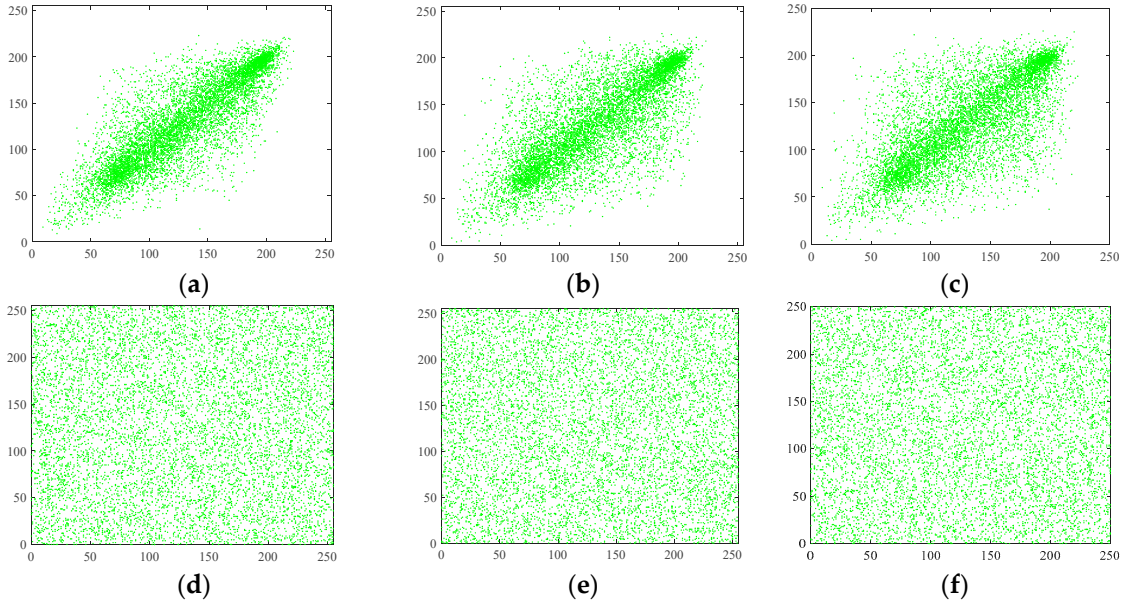


Figure 19. Correlation of the adjacent pixels on the G component of the Baboon color image: (**a**) Baboon—G—Horizontal; (**b**) Baboon—G—Vertical; (**c**) Baboon—G—Diagonal; (**d**) Encrypted—G—Horizontal; (**e**) Encrypted—G—Vertical; (**f**) Encrypted—G—Diagonal.

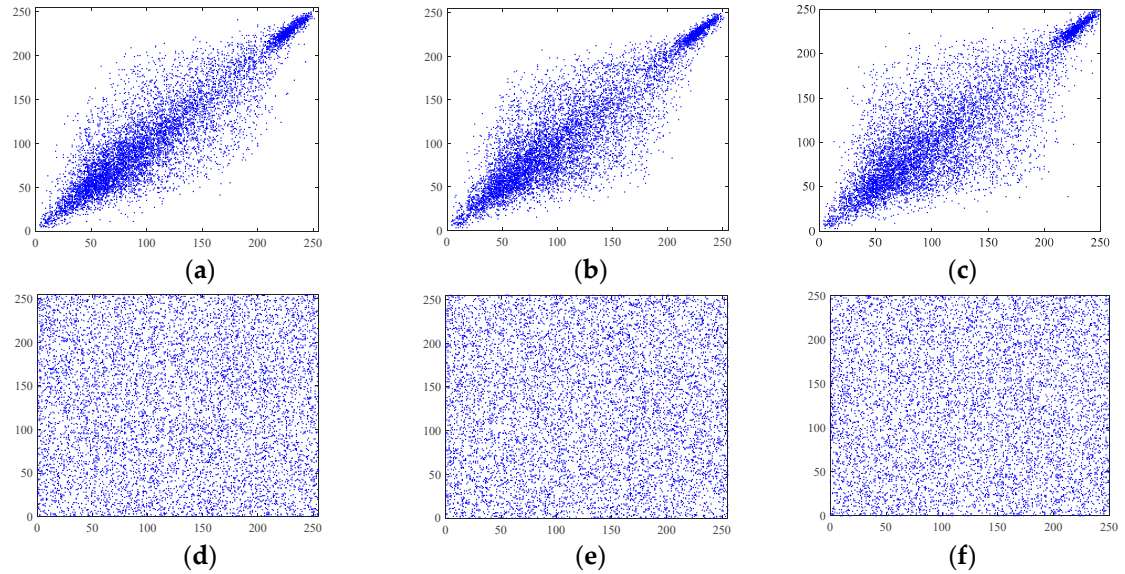


Figure 20. Correlation of the adjacent pixels on the B component of the Baboon color image: (**a**) Baboon—B—Horizontal; (**b**) Baboon—B—Vertical; (**c**) Baboon—B—Diagonal; (**d**) Encrypted—B—Horizontal; (**e**) Encrypted—B—Vertical; (**f**) Encrypted—B—Diagonal.

5.7. Occlusion Attack Analysis

To evaluate the robustness of resisting occlusion attacks [55], 10%, 25%, and 50% of Peppers images are deleted, as drawn in Figure 21a–c. The decryption images are drawn in Figure 21d–f. We can clearly see that the decrypted images are almost the same as the plain images. Therefore, the proposed algorithm can resist data cropping attacks.

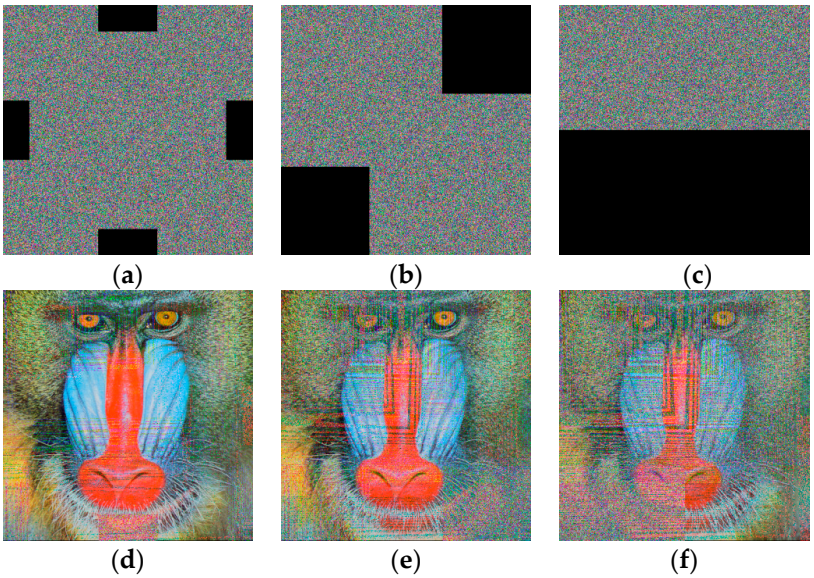


Figure 21. Simulation results of the occlusion attack: (**a**) 10% occlusion; (**b**) 25% occlusion; (**c**) 50% occlusion; (**d**) Decryption image of (**a**); (**e**) Decryption image of (**b**); (**f**) Decryption image of (**c**).

5.8. Chosen-Plaintext Attack

This is example 1 of an equation:

In an encryption system, the attack is very threatening to the encryption system. Therefore, an image encryption system must have enough strength to resist the attack analysis [56]. The result in Figure 22 indicates the encrypted images for three special images, i.e., the all-white image, all-black image, all-purple image. Therefore, the plain image features are corrupted. The proposed algorithm is reliable.

5.9. Randomness Test

The NIST test includes 15 tests [57]. The *p*-values of each test are calculated, and they should be greater than 0.01 in the test. The encrypted image of the Baboon is tested. A total of 100 repeat tests each have about 6.3 million bits. Table 8 shows the NIST test results of the Baboon, and we can find that the encrypted image of the Baboon has passed all the random tests. Therefore, the encryption effect of the proposed algorithm is excellent.

Table 8. The NIST test.

Test	*p*-Values	Pass/File
Random excursions variant test	0.9921	Passed
Frequency test	0.7652	Passed
Frequency test within a block	0.0975	Passed
Runs test	0.8743	Passed
Test for the longest run of the ones in a block	0.0871	Passed
Binary matrix rank test	0.4563	Passed
Discrete Fourier transform test	0.7611	Passed
Non-overlapping template matching test	0.2187	Passed
Overlapping template matching test	0.3125	Passed
Maurer's "Universal Statistical" test	0.5692	Passed
Linear complexity test	0.1143	Passed
Serial test	0.3217	Passed
Approximate entropy test	0.5689	Passed
Cumulative sums test	0.4303	Passed
Random excursions test	0.7615	Passed

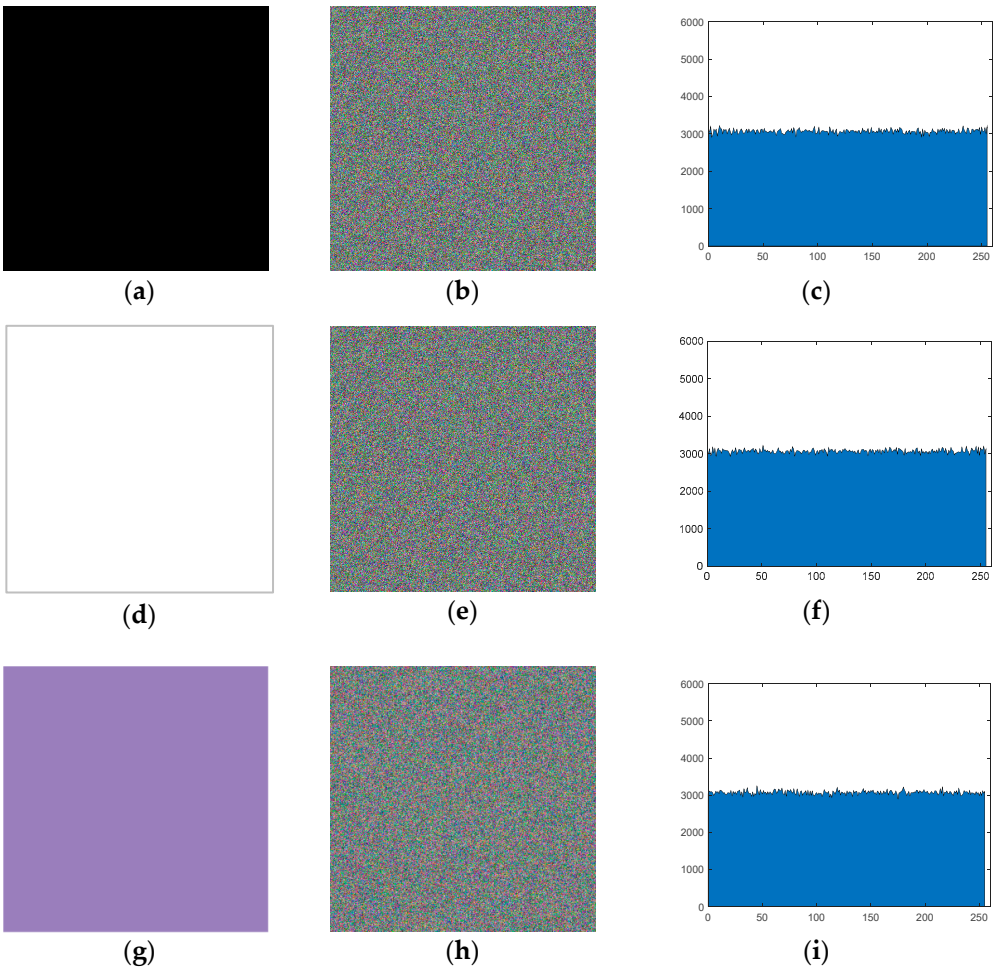


Figure 22. Tests for the chosen-plaintext attacks: (**a**) All-black image; (**b**) Encrypted image of (**a**); (**c**) Histogram of (**b**); (**d**) All-white image; (**e**) Encrypted image of (**d**); (**f**) Histogram of (**e**); (**g**) All-purple image; (**h**) Encrypted image of (**g**); (**i**) Histogram of (**h**).

5.10. Encryption Time and Computational Complexity Analysis

The proposed algorithm is suitable for color images of any size, and its computational complexity depends on the size of the plain image. Firstly, in the chaotic sequence generation stage, the complexity of the sequence generated by Chen's map, and the PWLCM map is about $O(3 \times m \times n)$. Secondly, in the scrambling stage, it is mainly reflected in the cross-spiral transformation of the three planes of the color image, and its complexity is $O(m \times n)$. Finally, in the diffusion stage, it is mainly reflected in the bit-level partitioned diffusion, and its complexity is $O(8 \times m \times n)$. In summary, the total complexity of the proposed algorithm is about $O(m \times n)$. Therefore, the higher the resolution of the image, the higher the complexity.

The encrypted time is a parameter that affects the feasibility of an encryption algorithm. Efficiency becomes especially important when encryption systems reach a certain level of security. In the experiment, color images sizes of $256 \times 256 \times 3$ and $512 \times 512 \times 3$ were tested several times, and their average was calculated.

Table 9 shows the computational complexity and encrypted time with other algorithms under the same image. It can be seen that the larger the plain image size, the higher the computational complexity of the proposed algorithm, and the encryption speed of the proposed algorithm is relatively fast and not inferior to other algorithms. Therefore, the proposed algorithm is efficient and suitable for real-world scenarios of image transmission.

Table 9. Computational complexity and encryption time analysis.

Algorithms	Size	Resolution	Time	Simulation Software
Proposed	256 × 256 × 3	256 × 256	0.5 s	MATLAB
	512 × 512 × 3	512 × 512	1.7 s	
	1024 × 1024 × 3	1024 × 1024	3.1 s	
Ref. [22]	256 × 256 × 3	256 × 256	1.1 s	MATLAB
Ref. [44]	512 × 512 × 3	512 × 512	2.5 s	MATLAB
Ref. [45]	512 × 512 × 3	512 × 512	2.1 s	MATLAB
Ref. [46]	512 × 512 × 3	512 × 512	1.7 s	MATLAB
Ref. [58]	512 × 512	512 × 512	5.78 s	FPGA
Ref. [59]	512 × 512 × 3	512 × 512	5.18 s	FPGA

6. Conclusions and Outlooks

To improve efficiency and security, this paper designs a cross-spiral transformation and partition diffusion and proposes a new color image encryption algorithm. On the one hand, with the help of the characteristics of color images, the cross-spiral transformation is constructed; that is, the pixel values of the R, G, and B planes are randomly exchanged by the index matrix to change pixel position. On the other hand, the zone diffusion operation is used to change the pixel values, and high security is obtained. After analysis, it is found that the histogram of the encrypted image shows that the pixel distribution is uniform, the information entropy is close to the theoretical value, the key space is sufficient to resist brute force attacks, and the encryption speed is fast. It can be confirmed that the proposed algorithm can successfully encrypt color images with high security and effectively resist various illegal attacks. These analyses indicate the superiority of the proposed algorithm.

However, with the large-scale application of color images in various fields, the proposed color image encryption algorithms are only suitable for single-color images of any size. Multiple color image encryption algorithms will be studied to ensure their secure and efficient transmission in the future.

Author Contributions: Methodology, X.Z.; Software, M.L. and X.Y.; Validation, X.Y.; Data curation, X.Y.; Writing—original draft, M.L.; Writing—review & editing, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data generated and/or analyzed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Wang, C.; Wang, X.; Xia, Z.; Ma, B.; Shi, Y.Q. Image description with polar harmonic fourier moments. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 40–52. [CrossRef]
2. Asgari, C.M. A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation. *Signal Process.* **2019**, *157*, 1–13. [CrossRef]
3. Xiong, L.; Han, X. Robust reversible watermarking in encrypted image with secure multi-party based on lightweight cryptography. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 75–91. [CrossRef]
4. Demirtas, M. A novel multiple grayscale image encryption method based on 3D bit-scrambling and diffusion. *Optik* **2022**, *266*, 169624. [CrossRef]
5. Tong, L.; Zhou, N.; Huang, Z.; Xie, X.-W.; Liang, Y.-R. Nonlinear multi-image encryption scheme with the reality-preserving discrete fractional angular transform and DNA sequences. *Secur. Commun. Netw.* **2021**, *20*, 6650515. [CrossRef]
6. Zhu, S.; Deng, X.; Zhang, W.; Zhu, C. Image encryption scheme based on newly designed chaotic map and parallel DNA coding. *Mathematics* **2023**, *11*, 231. [CrossRef]
7. Huang, L.; Chai, B.; Xiang, J.; Zhang, Z.; Liu, J. Chaotic image encryption based on spiral traversal and finite field bidirectional diffusion. *Phys. Scr.* **2023**, *98*, 035217. [CrossRef]

8. Zhou, S.; Qiu, Y.; Wang, X.; Zhang, Y. Novel image cryptosystem based on new 2D hyperchaotic map and dynamical chaotic S-box. *Nonlinear Dyn.* **2023**, *111*, 9571–9589. [CrossRef]
9. Man, X.; Song, Y. Encryption of Color Images with an evolutionary framework controlled by chaotic systems. *Entropy* **2023**, *25*, 631. [CrossRef]
10. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1245–1257. [CrossRef]
11. Wang, X.; Guan, N.; Yang, J. Image encryption algorithm with random scrambling based on one-dimensional logistic self-embedding chaotic map. *Chaos Solitons Fractals* **2021**, *150*, 111–129. [CrossRef]
12. Naskar, P.K.; Bhattacharyya, S.; Mahatab, K.C.; Dhal, K.G.; Chaudhuri, A. An efficient block-level image encryption scheme based on multi-chaotic maps with DNA encoding. *Nonlinear Dyn.* **2021**, *105*, 3673–3698. [CrossRef]
13. Chen, S.; Lü, J. Parameters identification and synchronization of chaotic systems based upon adaptive control. *Phys. Lett. A* **2002**, *299*, 353–358. [CrossRef]
14. Rehman, A.U.; Liao, X. Selective encryption for gray images based on chaos and DNA complementary rules. *Multimed. Tools Appl.* **2015**, *74*, 4655–4677. [CrossRef]
15. Xian, Y.; Wang, X.; Yan, X.; Li, Q.; Wang, X. Image encryption based on chaotic sub-block scrambling and chaotic digit Selection diffusion. *Opt. Lasers Eng.* **2020**, *134*, 106202. [CrossRef]
16. Li, S.; Zhao, L.; Yang, N. Medical image encryption based on 2D Zigzag confusion and dynamic diffusion. *Secur. Commun. Netw.* **2021**, *2021*, 6624809. [CrossRef]
17. Lone, M.A.; Qureshi, S. RGB image encryption based on symmetric keys using Arnold transform, 3D chaotic map and affine hill cipher. *Optik* **2022**, *260*, 168880. [CrossRef]
18. Zhao, Y.; Meng, R.; Zhang, Y.; Yang, Q. Image encryption algorithm based on a new chaotic system with Rubik's cube transform and Brownian motion model. *Optik* **2023**, *273*, 170342. [CrossRef]
19. Shen, H.; Shan, X.; Xu, M.; Tian, Z. A new chaotic image encryption algorithm based on transversals in a latin square. *Entropy* **2022**, *24*, 1574. [CrossRef]
20. Tang, Z.; Yang, Y.; Xu, S.; Yu, C.; Zhang, X. Image encryption with double spiral scans and chaotic maps. *Secur. Commun. Netw.* **2019**, *2019*, 8694678. [CrossRef]
21. Yuan, H.; Jiang, L. Image scrambling based on spiral filling of bits. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2015**, *8*, 225–234. [CrossRef]
22. Wang, Q.; Zhang, X.; Zhao, X. Color image encryption algorithm based on bidirectional spiral transformation and DNA coding. *Phys. Scr.* **2023**, *98*, 25211. [CrossRef]
23. Dhiveyaswathi, T.; Balamurugan, G. An enhanced image encryption approach using four dimension hyperchaotic chen map. In Proceedings of the 2021 5th International Conference on Computer, Communication and Signal Processing, Chennai, India, 24–25 May 2021; pp. 89–93.
24. Xiao, Y.; Chen, Y.; Long, C.; Shi, J.; Ma, J.; He, J. A novel hybrid secure method based on DNA encoding encryption and spiral scrambling in chaotic OFDM-PON. *IEEE Photonics J.* **2020**, *12*, 1–15. [CrossRef]
25. Wang, X.; Chen, S. Chaotic image encryption algorithm based on dynamic spiral scrambling transform and Deoxyribonucleic Acid encoding operation. *Mathematics* **2020**, *8*, 160897–160914. [CrossRef]
26. Liu, Y.; Shen, X.; Liu, J.; Peng, K. Optical asymmetric JTC cryptosystem based on multiplication-division operation and RSA algorithm. *Opt. Laser Technol.* **2023**, *160*, 109042. [CrossRef]
27. Xian, Y.; Wang, X.; Wang, X.; Li, Q.; Yan, X. Spiral-transform-based fractal sorting matrix for chaotic image encryption. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 3320–3327. [CrossRef]
28. Xu, J.; Zhao, B. Designing an image encryption algorithm based on hyperchaotic system and DCT. *Int. J. Bifurc. Chaos* **2023**, *32*, 2350021. [CrossRef]
29. Wang, X.; Wang, X.; Teng, L.; Jiang, D.H.; Xian, Y. Lossless embedding: A visually meaningful image encryption algorithm based on hyperchaos and compressive sensing. *Chin. Phys. B* **2023**, *32*, 20503. [CrossRef]
30. Huang, H.; Yang, S. Color image encryption based on logistic mapping and double random-phase encoding. *IET Image Process* **2017**, *11*, 211–216. [CrossRef]
31. Zhu, H.; Dai, L.; Liu, Y.; Wu, L. A three-dimensional bit-level image encryption algorithm with Rubik's cube method. *Math. Comput. Simul.* **2021**, *185*, 754–770. [CrossRef]
32. Zhang, X.; Liu, Z.; Yang, X. Fast image encryption algorithm based on 2D-FCSM and pseudo-wavelet transform. *Nonlinear Dyn.* **2023**, *111*, 6839–6853. [CrossRef]
33. Xu, L.; Li, Z.; Li, J.; Hua, W. A novel bit-level image encryption algorithm based on chaotic maps. *Opt. Lasers Eng.* **2016**, *78*, 17–25. [CrossRef]
34. Wang, M.; Liu, H.; Zhao, M. Bit-level image encryption algorithm based on random-time S-Box substitution. *Eur. Phys. J. Spec. Top.* **2022**, *231*, 3225–3237. [CrossRef]
35. Wang, M.; Wang, X.; Zhang, Y.; Zhou, S.; Zhao, T.; Yao, N. A novel chaotic system and its application in a color image cryptosystem. *Opt. Lasers Eng.* **2019**, *121*, 479–494. [CrossRef]
36. Zhang, Q.; Han, J. A novel color image encryption algorithm based on image hashing, 6D hyperchaotic and DNA coding. *Multimed. Tools Appl.* **2021**, *80*, 13841–13864. [CrossRef]

37. Liu, H.; Jin, C. A color image encryption scheme based on arnold scrambling and quantum chaotic. *Int. J. Netw. Secur.* **2017**, *19*, 347–357.
38. Hu, C.; Xie, X.; Zhou, N. Colour image encryption scheme based on the real-valued discrete Gabor transform. *J. Mod. Opt.* **2022**, *69*, 511–522. [CrossRef]
39. Gan, Z.; Chai, X.; Han, D.; Chen, Y.-R. A chaotic image encryption algorithm based on 3-D bit-plane permutation. *Neural Comput. Appl.* **2019**, *31*, 7111–7130. [CrossRef]
40. Lu, J.; Chen, G. A new chaotic attractor coined. *Int. J. Bifurc. Chaos* **2002**, *12*, 659–661. [CrossRef]
41. Tian, J.; Lu, Y.; Zuo, X.; Liu, Y.; Qiao, B.; Fan, M.; Ge, Q.; Fan, S. A novel image encryption algorithm using PWLCM map-based CML chaotic system and dynamic DNA encryption. *Multimed. Tools Appl.* **2021**, *80*, 32841–32861. [CrossRef]
42. The University of Southern California SIPI Image Database. Available online: http://sipi.usc.edu/database (accessed on 5 April 2023).
43. Ahmad, P.K.; Ahmad, H.N.; Massoud, B.A.; Mirnia, M. A novel multi-image cryptosystem based on weighted plain images and using combined chaotic maps. *Multimed. Syst.* **2021**, *27*, 907–925.
44. Ashish, G.; Vijay, K. A RGB image encryption technique using Lorenz and Rossler chaotic system on DNA sequences. *Multimed. Tools Appl.* **2018**, *77*, 27017–27039.
45. Li, T.; Shi, J.; Zhang, D. Color image encryption based on joint permutation and diffusion. *J. Electron. Imaging* **2021**, *30*, 13008. [CrossRef]
46. Teng, L.; Wang, X.; Yang, F.; Xian, Y. Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion. *Nonlinear Dyn.* **2021**, *105*, 1859–1876. [CrossRef]
47. Su, Q.; Zhang, X.; Wang, H. A blind color image watermarking algorithm combined spatial domain and SVD. *Int. J. Intell. Syst.* **2021**, *37*, 4747–4771. [CrossRef]
48. Muhammad, A.; Tabasam, R.; Sohail, Z. An image encryption scheme proposed by modifying chaotic tent map using fuzzy numbers. *Multimed. Tools Appl.* **2022**, *10*, 16861–16879.
49. Zhang, X.; Gao, T. Multiple-image encryption algorithm based on the bit plane and superpixel. *Multimed. Tools Appl.* **2022**, *12*, 19969–19991. [CrossRef]
50. Zhang, X.; Gong, Z. Color image encryption algorithm based on 3D Zigzag transformation and view planes. *Multimed. Tools Appl.* **2022**, *81*, 31753–31785. [CrossRef]
51. Zhang, Y.; Xie, H.; Sun, J.; Zhang, H. An efficient multi-level encryption scheme for stereoscopic medical images based on coupled chaotic system and Otsu threshold segmentation. *Comput. Biol. Med.* **2022**, *14*, 105542. [CrossRef]
52. Ahmad, L.M.; Shaima, Q. Encryption scheme for RGB images using chaos and affine hill cipher technique. *Nonlinear Dyn.* **2023**, *111*, 5919–5939.
53. Erkan, U.; Toktas, A.; Toktas, F.; Alenezi, F. 2D eπ-map for image encryption. *Inf. Sci.* **2022**, *589*, 770–789. [CrossRef]
54. Wen, J.; Xu, X.; Sun, K.; Jiang, Z.; Wang, X. Triple-image bit-level encryption algorithm based on double cross 2D hyperchaotic map. *Nonlinear Dyn.* **2023**, *111*, 6813–6838. [CrossRef]
55. Zhang, X.; Hu, Y. Multiple-image encryption algorithm based on the 3D scrambling model and dynamic DNA coding. *Opt. Laser Technol.* **2021**, *141*, 107073. [CrossRef]
56. Zhou, S.; Wang, X.; Zhang, Y. Novel image encryption scheme based on chaotic signals with finite-precision error. *Inf. Sci.* **2023**, *62*, 782–798. [CrossRef]
57. Zhang, X.; Liu, M.; Tian, J.; Gong, Z. Color image encryption algorithm based on dynamic block Zigzag transformation and six-sided star model. *Electronics* **2022**, *11*, 2512. [CrossRef]
58. Yu, F.; Xu, S.; Xiao, X.; Yao, W.; Huang, Y.; Cai, S.; Yin, B.; Li, Y. Dynamics analysis, FPGA realization and image encryption application of a 5D memristive exponential hyperchaotic system. *Integration* **2023**, *90*, 58–70. [CrossRef]
59. Doubla, I.S.; Njitacke, Z.T.; Ekonde, S.; Tsafack, N.; Nkapkop, J.D.D.; Kengne, J. Multistability and circuit implementation of tabu learning two-neuron model: Application to secure biomedical images in IoMT. *Neural Comput. Appl.* **2021**, *33*, 14945–14973. [CrossRef]