fractal and fractional

MDPI

Article

A Color Image-Encryption Algorithm Using Extended DNA Coding and Zig-Zag Transform Based on a Fractional-Order Laser System

Fanqi Meng * and Zhenglan Gu

School of Mathematics and Statistics, Yancheng Teachers University, No. 50, Kaifang Avenue, Yancheng 224002, China; zhenglangu@126.com
* Correspondence: fqmeng@hhu.edu.cn

Abstract: With the advancement of information technology, the security of digital images has become increasingly important. To ensure the integrity of images, a novel color image-encryption algorithm based on extended DNA coding, Zig-Zag transform, and a fractional-order laser system is proposed in this paper. First, the dynamic characteristics of the fractional-order laser chaotic system (FLCS) were analyzed using a phase diagram and Lyapunov exponent spectra. The chaotic sequences generated by the system were used to design image-encryption algorithms. Second, a modified Zig-Zag confusing method was adopted to confuse the image. Finally, in the diffusion link, the DNA encoding scheme was extended to allow for a greater number of DNA encoding rules, increasing the randomness of the matrix and improving the security of the encryption scheme. The performance of the designed encryption algorithm is analyzed using key space, a histogram, information entropy, correlation coefficients, differential attack, and robustness analysis. The experimental results demonstrate that the algorithm can withstand multiple decryption methods and has strong encryption capability. The proposed novel color image-encryption scheme enables secure communication of digital images.

Keywords: image encryption; fractional-order laser chaotic system; Zig-Zag transform; extended DNA coding

Citation: Meng, F.; Gu, Z. A Color Image-Encryption Algorithm Using Extended DNA Coding and Zig-Zag Transform Based on a Fractional-Order Laser System. *Fractal Fract.* **2023**, *7*, 795. https://doi.org/10.3390/fractalfract7110795

Academic Editor: Viorel-Puiu Paun

Received: 8 September 2023
Revised: 12 October 2023
Accepted: 24 October 2023
Published: 31 October 2023

1. Introduction

With advances in network communications technology, data can now be shared over open public networks and stored on a variety of platforms. As a result, ensuring the security and confidentiality of data has become critical. Any unauthorized access, appropriation, or destruction of network information can not only cause financial losses for computer users but also pose a major threat to the security of whole societies or even countries. Cryptography is a well-known technique for hiding secret information. In cryptography, images and text are encrypted before being transmitted over a network. The inherent characteristics of images, such as tight correlation, high redundancy, and block data capacity between adjacent pixels, distinguish image encryption from text encryption. Encryption is the process of hiding secret information by converting it into an unrecognizable form [1–5]. Therefore, network information security has become an important area of scientific research. Among all types of information in the form of data, images are the most important media for information exchange, and their security is particularly important.

Chaotic systems have been widely used in image encryption due to their excellent properties, such as ergodicity, pseudo-randomness, and sensitivity to system parameters and initial conditions. Ma et al. [6] investigated a fast encryption algorithm based on a 5D chaotic system. Qu et al. [7] proposed a color image-encryption method based on Hadamard single-pixel imaging and Arnold transform. Kumar et al. [8] applied the generalized heat equation for image encryption. Patro et al. [9,10] used chaotic maps to design an image-encryption algorithm. Khalid et al. [11] designed a color image-encryption algorithm based

on fractional shifted Gegenbauer moments and a 2D logistic sine map. Gao et al. [12] designed a color image-encryption algorithm using a hyperchaotic map. Arpaci et al. [13] studied a color image-encryption algorithm based on Chua's circuit. Yu et al. [14] designed an image-encryption application for the multiscroll memristive Hopfield neural network. Ren et al. [15] designed an image-encryption algorithm using a hyperchaotic map with a memristor. Laser-generated chaos can break the electronic bottleneck of traditional chaos, offering the advantages of high bandwidth, complexity, and propagation rates. These advantages have led to laser chaos systems' widespread use in the field of image encryption and confidential communication. Wang et al. [16] proposed a color image-encryption strategy based on a double-layer Josephus scramble and a laser chaotic system. Fractional-order calculus is an extension of integer-order calculus based on the advantages of fractional-order chaotic systems, which are closer to self-bounded theoretical properties and have more complex dynamics than integer-order chaotic systems [17–19]. Li et al. [20] designed an image-encryption algorithm based on a fractional-order laser hyperchaotic system. In this study, we designed image-encryption algorithms based on fractional-order laser systems due to their complex dynamics. Although some results have been achieved in the study of image encryption using chaotic systems, the security of individual image-encryption methods using only chaotic systems depends on the complexity of chaotic systems. Therefore, the combination of chaos theory with other theories should be explored to improve the security of image encryption.

The Zig-Zag transform is a simple and effective method of rearranging the pixels of an image. Essentially, this method consists of scanning the elements of a matrix in a specific order, called the Zig-Zag order, starting from the top left corner and moving toward the bottom right corner to scramble the data. The Zig-Zag transform has been widely used in the field of image and video encryption due to its simplicity and low time complexity. Guo et al. [21] proposed an image-encryption algorithm using the reverse Zig-Zag transformation method. Gao et al. [22] applied a dynamic Zig-Zag transform and row-scrambling method in image encryption. In addition, DNA computing has received extensive attention from domestic and foreign researchers due to its high parallelism, low energy consumption, and massive storage capacity. Digital image-encryption methods based on DNA computing entail the use of the rules of DNA coding to convert digital images into DNA sequences according to certain coding rules and then operate on them according to the rules of DNA computing [23–26]. More secure and efficient encryption algorithms can be designed by combining chaotic systems with DNA encoding techniques. Yildirim [27] designed a method for color image encryption based on chaotic circuits and extended DNA encoding. Wang et al. [28] presented an image-encryption method using DNA encoding and compressed sensing. Yan et al. [29] proposed an image-encryption algorithm based on DNA coding sequences and a 1D logistic map. Wang et al. [30] applied Fisher–Yates scrambling and a DNA subsequence operation to carry out image encryption. Zhang et al. [31] combined DNA encoding, phase-truncated FRFT, a hyperchaotic system, and Arnold transform to apply an image-encryption method. Based on the existing references in the literature, most of the image-encryption algorithms based on DNA coding are combined with integer-order chaotic systems, which are known to have richer dynamic characteristics due to their highly nonlinear and nonlocal properties [32–36].

It is worth noting that by observing the current technology, most algorithms for DNA coding are based on the four-gene model, and most encryption algorithms based on laser chaotic systems are integer-order models. However, there is not much research on encryption algorithms that combine extended DNA coding methods with a fractional-order laser system, constituting a very valuable direction of research. Therefore, the primary contribution of this paper is the design of a new Zig-Zag transform method, combined with the extension of a DNA encoding method from four bases to eight bases and a fractional-order laser system, leading to the development of a new image-encryption algorithm. The main highlights of this work are summarized as follows.

(1) A block Zig-Zag transform method is designed to increase the complexity of image scrambling.
(2) The nonlinear dynamical characteristics of FLCS are analyzed using bifurcation diagrams, Lyapunov exponents, and phase diagrams.
(3) Based on the sensitivity of the initial value of FLCS and the complex chaotic dynamics properties, a new color image-encryption algorithm based on block Zig-Zag transform, extended DNA coding, and FLCS is designed.
(4) The comprehensive performance analysis and statistical analysis results show that the proposed encryption algorithm is highly secure.

This paper is structured as follows: Section 2 outlines the preliminary material and mathematical models used in this study. Section 3 gives the detailed procedure of the color image-encryption algorithm based on block Zig-Zag transform and extended DNA coding combined with FLCS. The experimental simulation results are described in Section 4. Section 5 verifies the comprehensive performance analysis of the proposed encryption algorithm. Finally, some conclusions are given in Section 6.

2. Preliminary Materials

This section analyses the dynamics characteristics of FLCS and designs the block Zig-Zag transform method for later investigation of color image encryption and decryption algorithm.

2.1. Fractional-Order 4D Chaotic Laser System

The four-dimensional chaotic laser system derives from the famous Lorenz–Haken equations [37], which can be expressed as:

$$\begin{cases} \dot{x} = -\sigma(y-x) + iqx|x|^2 \\ \dot{y} = -(1-i\delta)y + (r-z)x \\ \dot{z} = -bz + Re(x*y) \end{cases}, \tag{1}$$

where x is proportional to the electric field, y is proportional to the induced macroscopic polarization, τ_N denotes the inversion parameter, τ_P is the induced polarization, τ_E represents the optical field, $\sigma = \tau_P/\tau_E$, $b = \tau_P/\tau_N$, $(r-z)$ denotes the inversion. Meanwhile, the parameter q is known as the linewidth enhancement factor, δ governs the coupling between phase variations and amplitude.

Since the x and z are chosen as real parts, the dynamics of the original laser equation are studied by the following linear transformation: $x = x_1$, $y = x_2 + ix_3$, $z = x_4$. The modified laser system is defined as:

$$\begin{cases} \dot{x_1} = \sigma(x_2 - x_1) \\ \dot{x_2} = -x_2 - \delta x_3 + (r - x_4)x_1 \\ \dot{x_3} = \delta x_2 - x_3 \\ \dot{x_4} = -bx_4 + x_1x_2 \end{cases}, \tag{2}$$

where σ, r, b, δ are system parameters, and x_i are system state variables.

Fractional calculus is an extension of integer calculus. Dynamic systems calculated using fractional differential equations have more complex dynamic properties, leading to the generation of more complicated chaotic sequences. The Caputo-type fractional-order differential equation is defined as:

$$D_t^q f(t) = \frac{1}{\Gamma(w-q)} \int_0^t \frac{f^w(\tau)}{(t-\tau)^{q-w+1}} d\tau, \tag{3}$$

where $\Gamma(\cdot)$ is the gamma function and $t \geq 0, w \in Z^+$, $w-1 < q < w$.

According to the definition of a fractional-order differential equation, the FLCS is defined in Equation (4).

$$\begin{cases} D_t^{\alpha} x_1(t) = \sigma(x_2(t) - x_1(t)) \\ D_t^{\alpha} x_2(t) = -x_2(t) - \delta x_3(t) + (r - x_4(t))x_1(t) \\ D_t^{\alpha} x_3(t) = \delta x_2(t) - x_3(t) \\ D_t^{\alpha} x_4(t) = -b x_4(t) + x_1(t)x_2(t) \end{cases}, \tag{4}$$

where α represents the order.

A Lyapunov exponent diagram, bifurcation diagram, and phase diagram of the time series of FLCS variables can indicate the chaotic dynamical behaviors of the system. We set the initial values and the parameters of FLCS as $(x_{10}, x_{20}, x_{30}, x_{40}) = (0.2, -0.1, 0.1, -0.2)$, $r = 18$, $b = 0.5$, $q = 0.97$. Figure 1a,b represent the bifurcation and Lyapunov exponential diagrams when the system parameters are $2 \leq \sigma \leq 6, \delta = 1.5$, respectively, and in Figure 1c,d are the diagrams when the system parameters $\sigma = 6, -2 \leq \delta \leq 2$. Figure 2 depicts its phase diagrams, in which the parameters are $\sigma = 2$, $\delta = 1.5$. FLCS exhibits complex chaotic dynamical behavior.

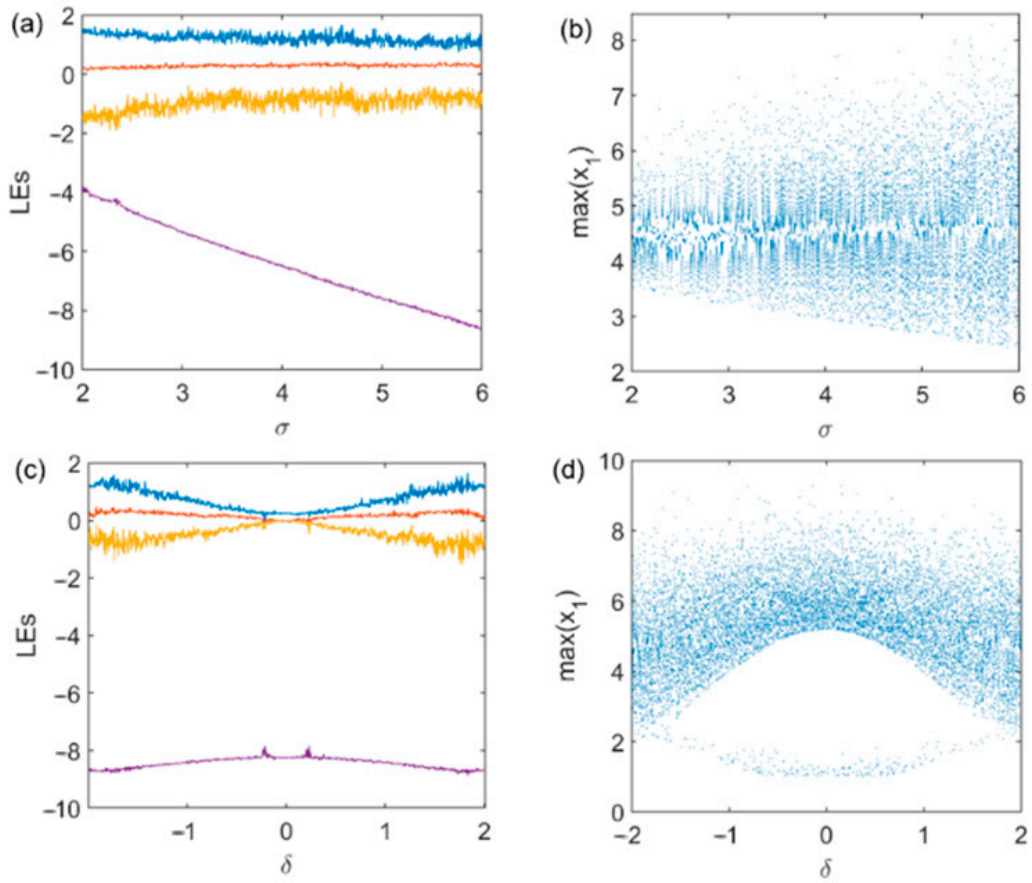


Figure 1. Bifurcation diagrams and Lyapunov exponent diagrams of FLCS. (**a**,**b**) are Lyapunov exponent and Bifurcation diagrams when $2 \leq \sigma \leq 6$, $\delta = 1.5$; (**c**,**d**) are Lyapunov exponent and Bifurcation diagrams when $\sigma = 6, -2 \leq \delta \leq 2$, colors navy, orange, yellow and magenta represent the Lyapunov exponent of variables x_1, x_2, x_3, x_4, respectively in (**a**,**c**), color navy represent the max value of variables x_1 in (**b**,**d**).

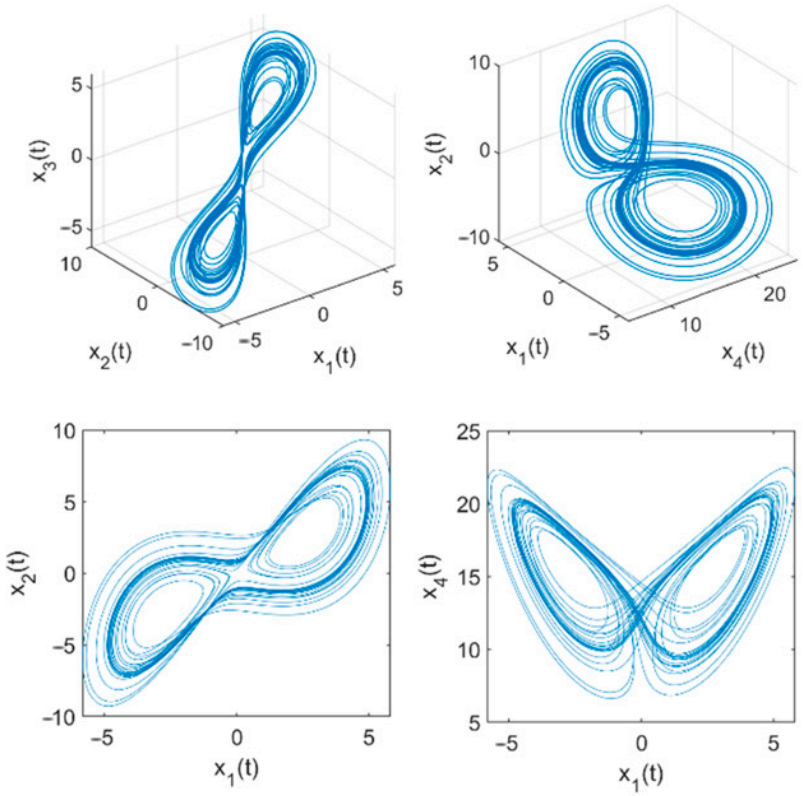


Figure 2. Phase diagrams of FLCS.

2.2. Block Zig-Zag Transform

The Zig-Zag transformation is a process by which the elements of a two-dimensional matrix are arranged in an alphabetical "Z" pattern and stored in a one-dimensional array. In image-processing techniques, each pixel is stored in a two-dimensional matrix, and then the Zig-Zag transformation is applied to the columns of the two-dimensional matrix. The result of the transformation is to update the image by shifting the original image. This paper extends the Zig-Zag transformation method for color image displacement. The specific extension method is shown in Figure 3.

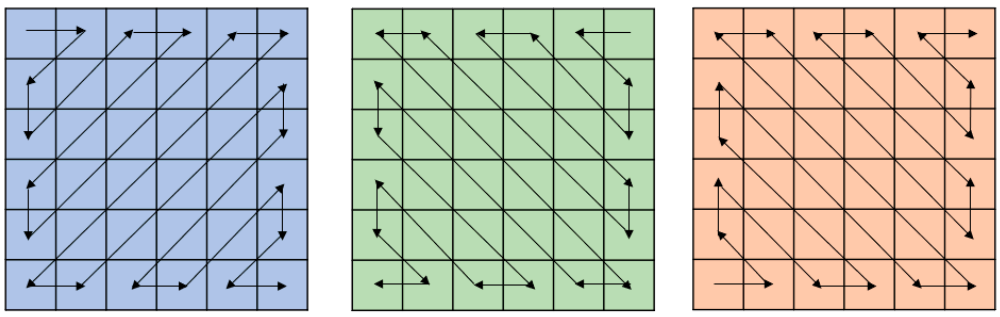


Figure 3. Block Zig-Zag Scrambling Method. (**a**) Upper left transform; (**b**) Upper right transform; (**c**) Lower left transform.

In the design of the image-encryption algorithm, the image is first divided and numbered according to equal blocks, and further Zig-Zag transformation is applied to different blocks based on chaotic sequences using the extended Zig-Zag method. The detailed image transformation process is shown in Figure 4.

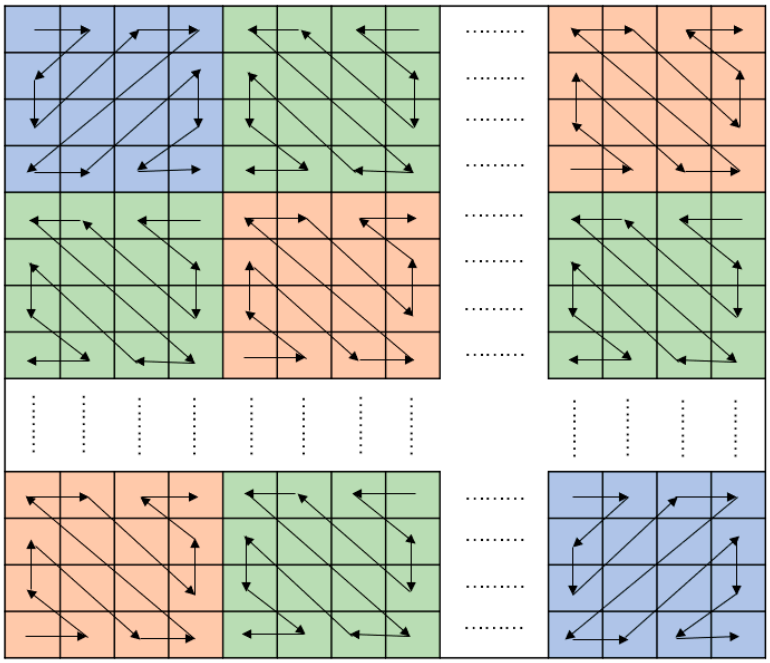


Figure 4. Block Zig-Zag transform scrambling process.

3. Image Encryption and Decryption Scheme

This section explains the 8-bit extended DNA coding and operation rules and details the process of color image encryption and decryption algorithms combined with the block Zig-Zag transform method and FLCS.

3.1. DNA Coding and Operations

The DNA sequence is made up of four bases: Adenine (A), Cytosine (C), Thymine (T), and Guanine (G). Adenine pairs with Thymine (A–T), and Cytosine pairs with Guanine (C–G). In this paper, four lowercase letters (a, t, c, and g) have been introduced to denote additional bases in addition to the original four uppercase letters. In the expanded representation, a is paired with t (a–t), and c is paired with g (c–g). To convert a color image into an extended DNA sequence for encoding and decoding, the following operations are performed. Then, a three-digit binary number in the form of a gene is used. Based on the complementary relationship between DNA bases, as the number of nucleotide bases increases from 4 to 8 bits, 384 of the 40,320 coding combinations allow complementary base pairing within the 8 bases. The coding rules for these 384 types are shown in Table 1.

Table 1. DNA coding rules.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	...	**384**
000	A	A	A	A	A	A	A	A	A	A	A	A	A	...	t
001	a	a	a	a	a	a	a	a	C	C	G	G	c	...	g
010	C	C	c	c	G	G	g	g	a	a	a	a	a	...	T
011	c	g	C	G	g	c	G	C	c	g	c	g	G	...	G
100	g	c	G	C	c	g	C	G	g	c	g	c	C	...	C
101	G	G	g	g	C	C	c	c	t	t	t	t	t	...	A
110	t	t	t	t	t	t	t	t	G	G	C	C	g	...	c
111	T	T	T	T	T	T	T	T	T	T	T	T	T	...	a

Each pixel of a color image is made up of three components, R, G, and B, each of which can be converted to 8 bits, and the three components combined to 24 bits. For example, a pixel of a color image consists of three components, C_R = 134, C_G = 120, and C_B = 38, which are combined and then coded in the 10th coding mode in Table 1, where

each component can be converted to 8 bits, and then the three components can be combined to 24 bits, C_RGB = [10000110, 01111000, 00100110]. By selecting the 10th encoding method in Table 1, C_RGB (10) = [c, C, c, T, c, A, c, G]. With this method, the color image can be converted into an extended DNA coding form.

The calculation of DNA sequences is achieved by bitwise operations, with four methods of bitwise operations, after encoding eight bases for addition, subtraction, XOR, and XNOR operations. Tables 2–5 show the four bitwise operations of addition, subtraction, XOR, and XNOR when [A, a, C, c, g, G, t, T] is represented by [000, 001, 010, 011, 100, 101, 110, 111], respectively.

Table 2. DNA addition operation.

+	**A**	**a**	**C**	**c**	**g**	**G**	**t**	**T**
T	T	A	a	C	c	g	G	t
t	t	T	A	a	C	c	g	G
G	G	t	T	A	a	C	c	g
g	g	G	t	T	A	a	C	c
c	c	g	G	t	T	A	a	C
C	C	c	g	G	t	T	A	a
a	a	C	c	g	G	t	T	A
A	A	a	C	c	g	G	t	T

Table 3. DNA subtraction operation.

−	**A**	**a**	**C**	**c**	**g**	**G**	**t**	**T**
T	T	t	G	g	c	C	a	A
t	t	G	g	c	C	a	A	T
G	G	g	c	C	a	A	T	t
g	g	c	C	a	A	T	t	G
c	c	C	a	A	T	t	G	g
C	C	a	A	T	t	G	g	c
a	a	A	T	t	G	g	c	C
A	A	T	t	G	g	c	C	a

Table 4. DNA XOR operation.

⊕	**A**	**a**	**C**	**c**	**g**	**G**	**t**	**T**
T	T	t	G	g	c	C	a	A
t	t	T	g	G	C	c	A	a
G	G	g	T	t	a	A	c	C
g	g	G	t	T	A	a	C	c
c	c	C	a	A	T	t	G	g
C	C	c	A	a	t	T	g	G
a	a	A	c	C	G	g	T	t
A	A	a	C	c	g	G	t	T

Table 5. DNA XNOR operation.

⊙	**A**	**a**	**C**	**c**	**g**	**G**	**t**	**T**
T	A	a	C	c	g	G	t	T
t	a	A	c	C	G	g	T	t
G	C	c	A	a	t	T	g	G
g	c	C	a	A	T	t	G	g
c	g	G	t	T	A	a	C	c
C	G	g	T	t	a	A	c	C
a	t	T	g	G	C	c	A	a
A	T	t	G	g	c	C	a	A

Based on the DNA encoding, decoding, and operation strategies, when the attacker tries to find the phase that matches the DNA encoding rules in the proposed algorithm, the probability of finding the correct encoding rule is 1/384 compared to the previous research results [28–31]. Additionally, DNA encoding and decoding operation rules such as addition, subtraction, XOR, and XNOR further enhance decoding possibilities to $(1/384)^4$. Therefore, the eight-base DNA coding method can be demonstrated to increase the security of image encryption.

3.2. Encryption Algorithm

The flowchart of the color image-encryption process based on Zig-Zag transformation, extended DNA coding, and FLCS is shown in Figure 5, and the detailed implementation steps of the encryption algorithm are as follows:

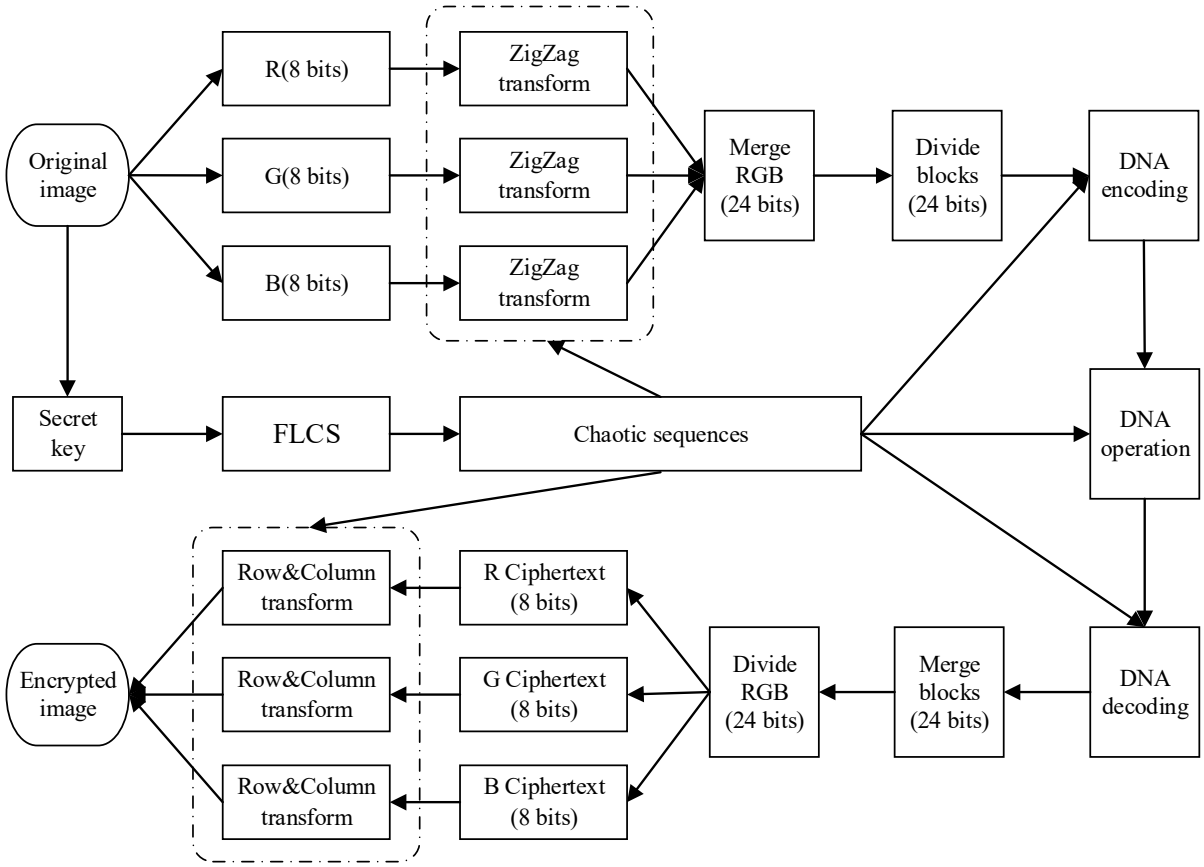


Figure 5. Flowchart of the encryption algorithm.

Step 1: Input the primitive plaintext image C of dimension $M \times N \times 3$ and divide the image into C_R, C_G, C_B components.

Step 2: Compute the initial value σ of FLCS as follows:

$$\sigma = 2 + round\left(\frac{Sum(C)}{M \times N \times 3 \times 255}, 8\right) \tag{5}$$

Step 3: Set the initial values x_{10}, x_{20}, x_{30}, x_{40}, r, b, q, σ, δ of FLCS as the secret key. The chaotic sequences Sx_1, Sx_2, Sx_3, Sx_4 are generated by FLCS.

Step 4: The chaotic sequences are calculated by Equation (6) to obtain Sx_z, Sx_e, Sx_o, Sx_d, Sx_b.

$$\begin{cases} Sx\_z = \left(round\left(|Sx_1| \times 10^8\right)\right) mod\ 3 \\ Sx\_e = \left(round\left(|Sx_1| \times 10^8\right)\right) mod\ 384 + 1 \\ Sx\_d = \left(round\left(|Sx_2| \times 10^8\right)\right) mod\ 384 + 1 \\ Sx\_o = \left(round\left(|Sx_3| \times 10^8\right)\right) mod\ 4 \\ Sx\_b = \left(round\left(|Sx_4| \times 10^8\right)\right) mod\ 256 \end{cases} \tag{6}$$

Step 5: The extended Zig-Zag transform is applied to the pixels in C_R, C_G, C_B using the three methods of chaotic sequence Sx_z representation, as shown in Figures 3 and 4, respectively.

Step 6: Merge the 8-bit binary pixel values of C_R, C_G, C_B into the 24-bit binary pixel values C_{RGB}.

Step 7: The chaotic sequence Sx_e represents the 384 DNA coding methods in Table 1. Divide C_{RGB} into equal blocks $C_{RGB}\ (i)$ and code each block separately according to Sx_e.

Step 8: The chaotic sequence Sx_4 is converted into equal chaotic block Sx_b and coded blocks, which are equal to $C_{RGB}\ (i)$.

Step 9: The chaotic sequence Sx_o represents the four DNA operations of addition, subtraction, XOR, and XNOR. The $C_{RGB}\ (i)$ is calculated with $Sx\_b(i)$ according to the DNA calculation operations defined by Sx_o to obtain $E_{RGB}\ (i)$.

Step 10: Merge matrix block $E_{RGB}\ (i)$ into E_{RGB} and decode E_{RGB} according to Sx_d.

Step 11: Convert 24-bit binary E_{RGB} to 8-bit binary E_R, E_G and E_B.

Step 12: Arrange x_1 and x_2 in descending order according to the following formula to obtain the sequence of positions Px_1 and Px_2 before the ordering of each element in the sequence.

$$\begin{cases} [\sim, Px_1] = sort(Sx_1(1:M)\ mod\ 1, 'descend') \\ [\sim, Px_2] = sort(Sx_2(1:N)\ mod\ 1, 'descend') \end{cases} \tag{7}$$

Perform row substitution and column substitution on the matrices of E_R, E_G, and E_B using the sequence values of Px_1 and Px_2 as the corresponding indices of the row and column exchange coordinates.

Step 13: Merge the R, G, and B components to obtain a ciphertext image.

3.3. Decryption Algorithm

Image decryption involves recovering the original image using the same key used for encryption. It is the inverse process of the image-encryption process. First, the R, G, and B components of the ciphertext image are inverted by row–column conversion, respectively, and the converted R, G, and B components are merged and divided into blocks. Second, encryption and decryption operations are performed on each piece of data using DNA operation rules that are the opposite of the encryption process. Then, the DNA-decoded data blocks are merged and divided into R, G, and B components. Each component is subjected to a Zig-Zag inverse transformation operation. Finally, the three components are merged into the initial image. The detailed flowchart of the decryption algorithm is presented in Figure 6.

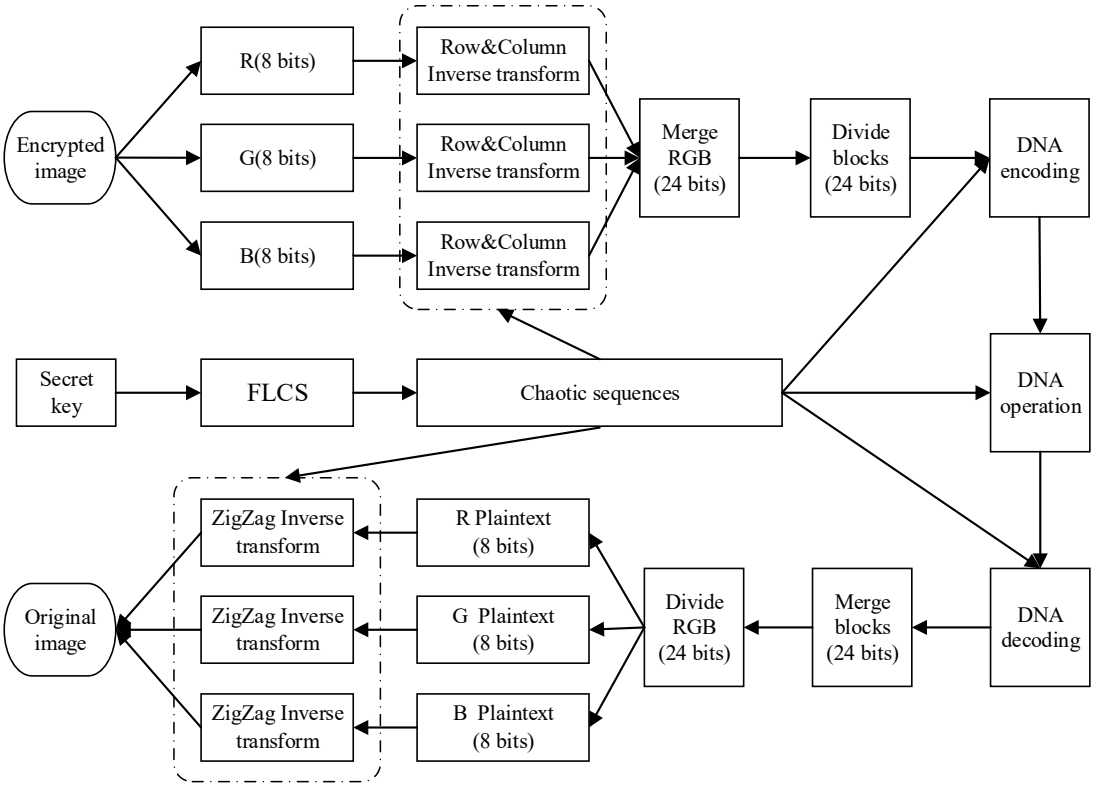
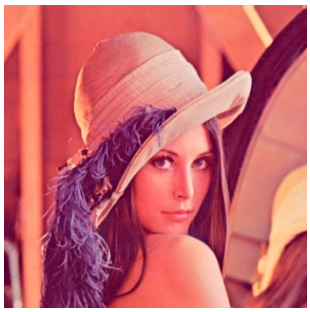


Figure 6. Flowchart of the decryption algorithm.

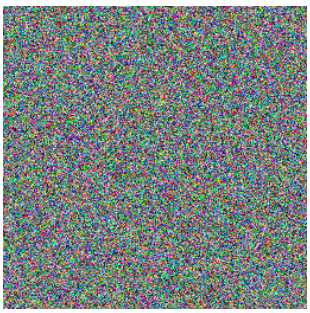
4. Experimental Simulation Results

In this paper, all the experiments are performed on a computer with an Intel Core i7-10710U 6-Core 1.10 Ghz and 32 GB of memory. The operating system is Windows 10, and the software is MATLAB R2020b. For the experiments and performance analyses, test images were selected from the web and publicly available databases such as the USC-SIPI database [38]. This section tests the effectiveness of encryption algorithms and decryption algorithms for different sizes and types of color images.

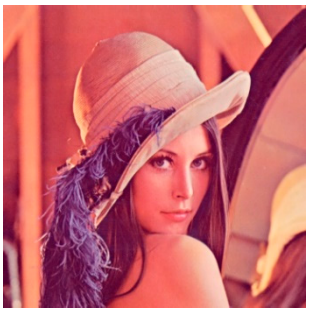
Lena image (Figure 7(a1), size 256 × 256), Fruits image (Figure 7(b1), size 512 × 480), Tree image (Figure 7(c1), size 256 × 256), and Peppers image (Figure 7(d1), size 512 × 512) were used to test the feasibility of the proposed algorithm. The secret key was set to $r = 18$, $b = 0.5$, $q = 0.97$, $\sigma = 2.5046$, $\delta = 1.5$, $x_{10} = 0.2$, $x_{20} = -0.1$, $x_{30} = 0.1$, $x_{40} = -0.2$. As can be seen from the experimental results, the encrypted images do not correlate with the original image, and there is no obvious difference between the reconstructed image and the original image visually, which indicates that the encryption scheme has produced good results.



(**a1**) Original Lena

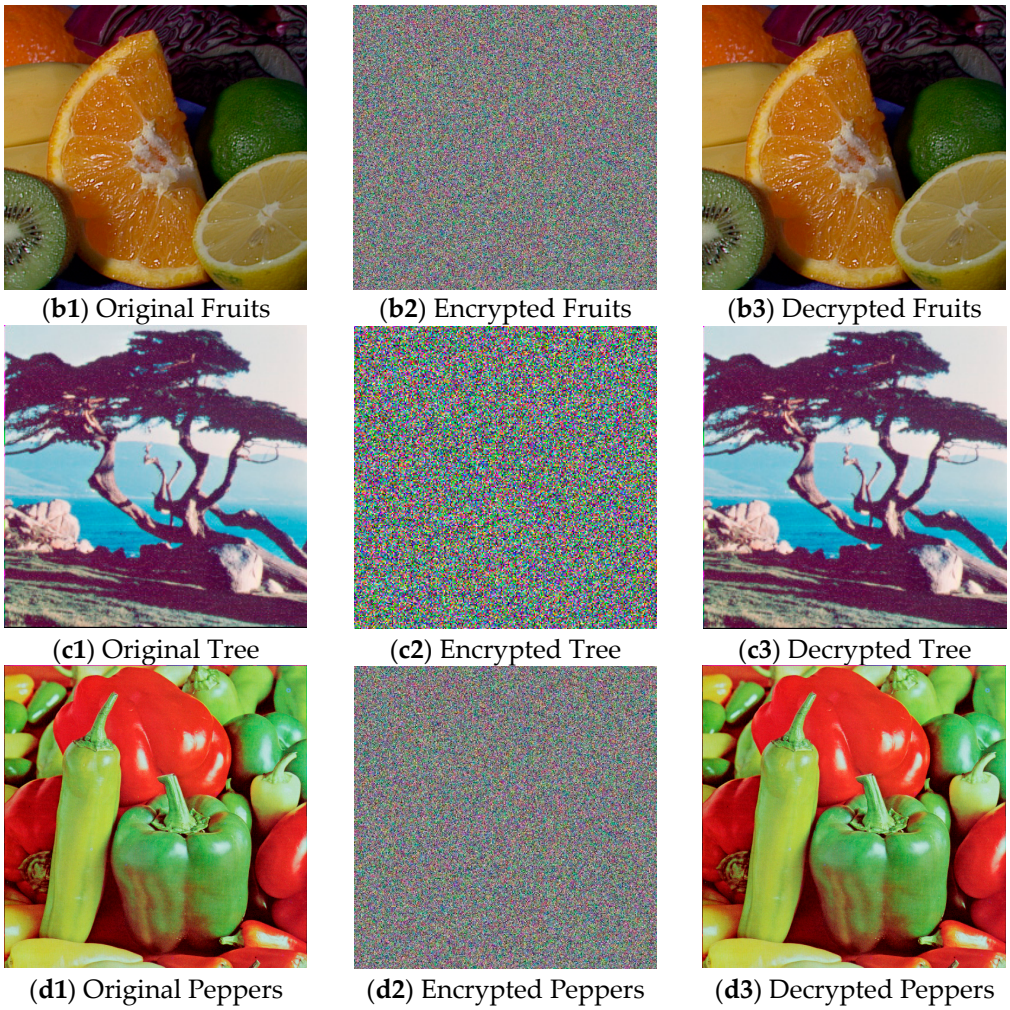


(**a2**) Encrypted Lena



(**a3**) Decrypted Lena

Figure 7. *Cont.*



(**b1**) Original Fruits (**b2**) Encrypted Fruits (**b3**) Decrypted Fruits

(**c1**) Original Tree (**c2**) Encrypted Tree (**c3**) Decrypted Tree

(**d1**) Original Peppers (**d2**) Encrypted Peppers (**d3**) Decrypted Peppers

Figure 7. Experimental results of encryption and decryption. (**a1**–**d1**) are the original images; (**a2**–**d2**) are the encrypted images; (**a3**–**d3**) are the decrypted images.

5. Performance Analyses

Good encryption algorithms require good security performance. This section mainly measures the security performance of the proposed algorithm using key security analysis, histogram analysis, correlation analysis, information entropy analysis, differential attack analysis, robustness analysis, classical attack analysis, and time complexity analysis. The experimental results show that the encryption algorithm designed in this paper can withstand various types of attacks.

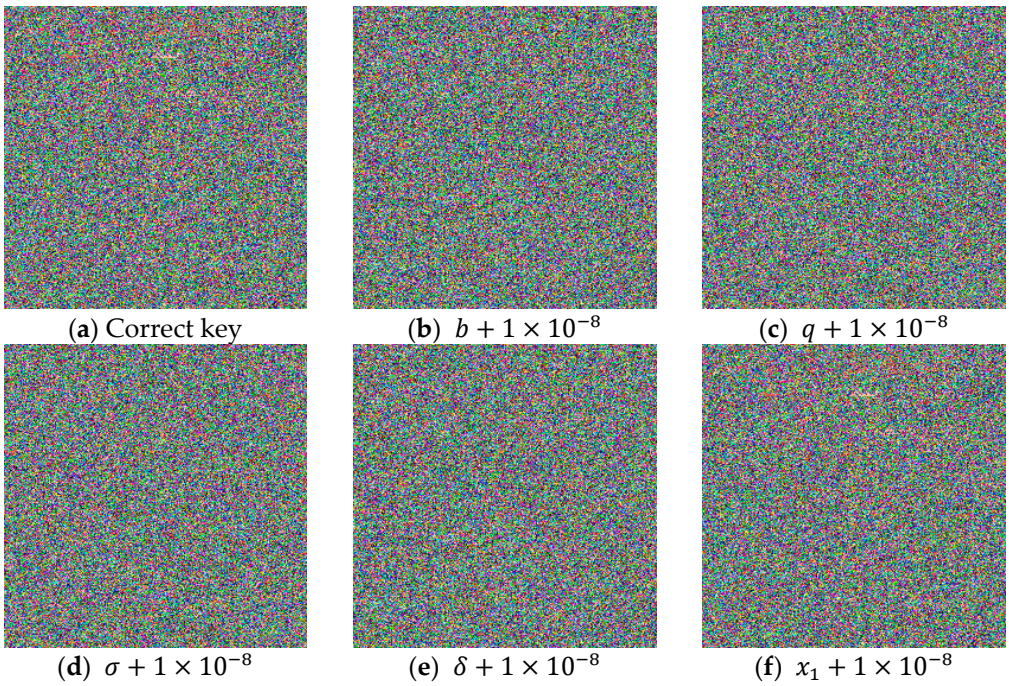
5.1. Key Analysis

Key analysis includes key space analysis and key sensitivity testing. The encryption algorithm designed in this paper uses four initial values and five parameters of FLCS as the secret key. Assuming that the computational accuracy of the computer is 10^{-8}, the secret key space is $10^{8\times 9} \approx 2^{239}$. This is a huge key capacity and is sufficient to resist the exhaustive key-based approach to cracking the image.

To test the key sensitivity, the Lena image (256×256) is adopted, each of five control parameters b, q, σ, δ, x_1 of FLCS is modified by adding 1×10^{-8} and only one parameter is changed at a time.

In the simulation test, the original image Lena is encrypted using both the correct and modified keys. The encrypted images are shown in Figure 8, and the ratio of pixel changes between the encrypted images using the correct and incorrect keys are calculated and listed

in Table 6. It is obvious that these encrypted images are dissimilar to each other, and more than 99.5% of the pixels have been changed.



(**a**) Correct key (**b**) $b + 1 \times 10^{-8}$ (**c**) $q + 1 \times 10^{-8}$

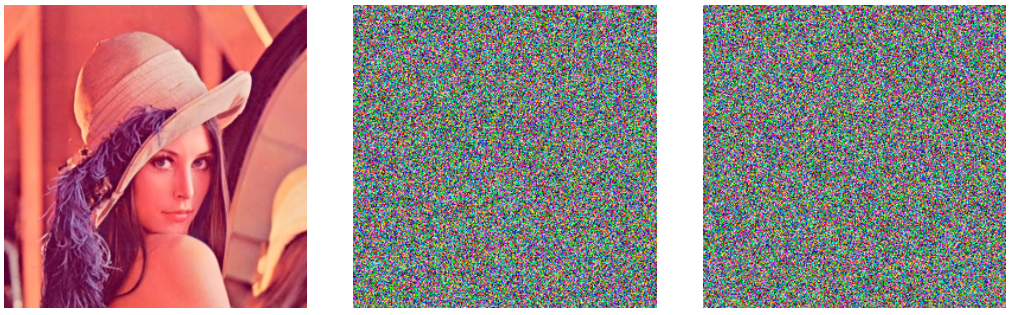
(**d**) $\sigma + 1 \times 10^{-8}$ (**e**) $\delta + 1 \times 10^{-8}$ (**f**) $x_1 + 1 \times 10^{-8}$

Figure 8. Key sensitivity test of encryption stage.

Table 6. Pixel change ratio between the correct and wrong keys.

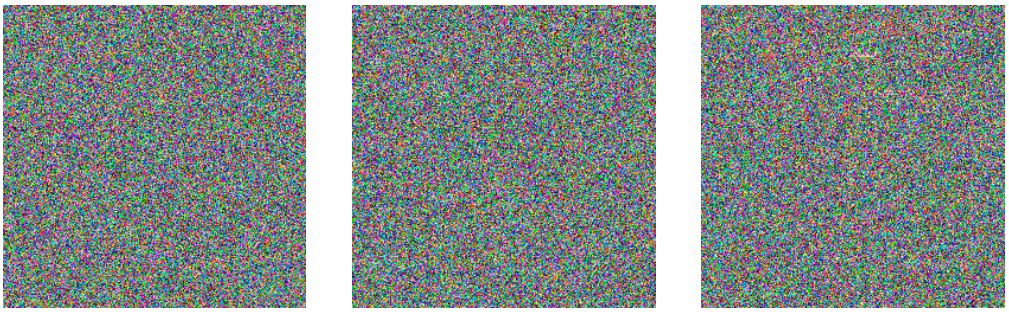
Encryption Key	Pixel Change Ratio			
	Red	**Green**	**Blue**	**Average**
Correct key	0%	0%	0%	0%
$b + 1 \times 10^{-8}$	99.635%	99.574%	99.609%	99.586%
$q + 1 \times 10^{-8}$	99.635%	99.66%	99.553%	99.624%
$\sigma + 1 \times 10^{-8}$	99.623%	99.608%	99.626%	99.614%
$\delta + 1 \times 10^{-8}$	99.577%	99.574%	99.664%	99.604%
$x_1 + 1 \times 10^{-8}$	99.594%	99.611%	99.623%	99.615%

In addition, decryption was performed with the correct key and the slightly incorrect keys to evaluate the key sensitivity of the decryption process. The experimental results in Figure 9 show that even with a slight change in the key, the correct plaintext image cannot be decrypted, indicating that the algorithm is highly sensitive to the key.



(**a**) Correct key (**b**) $b + 1 \times 10^{-8}$ (**c**) $q + 1 \times 10^{-8}$

Figure 9. *Cont.*

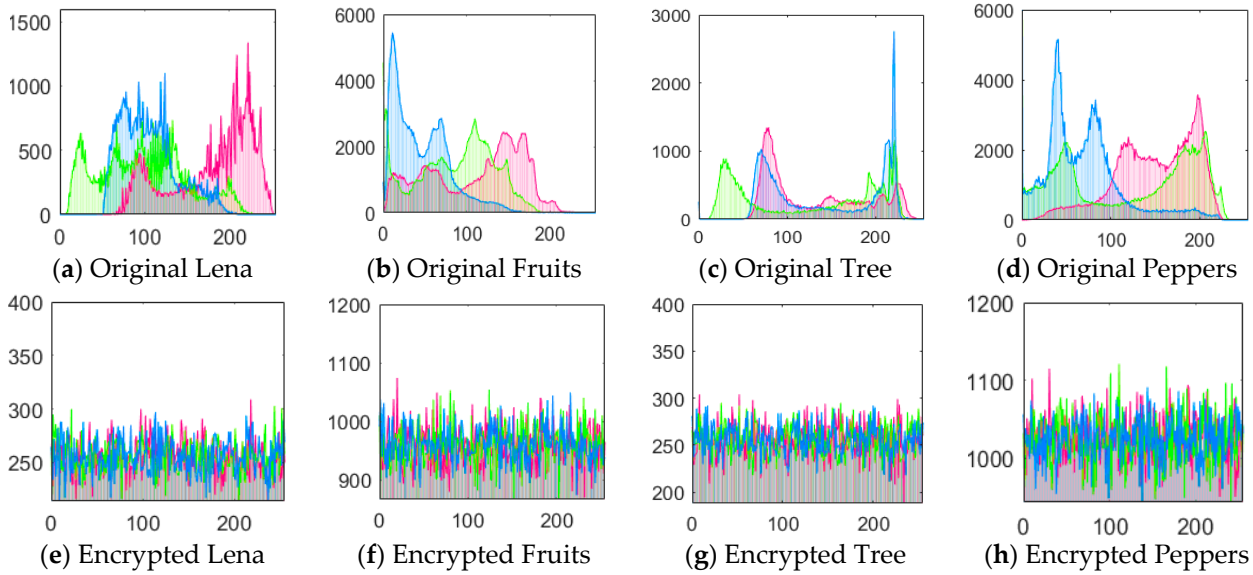


(**d**) $\sigma + 1 \times 10^{-8}$ (**e**) $\delta + 1 \times 10^{-8}$ (**f**) $x_1 + 1 \times 10^{-8}$

Figure 9. Key sensitivity test of decryption stage.

5.2. Histogram Analysis

The histogram of pixels reveals the distribution of all the pixels in an image. An efficient encryption algorithm can encrypt the pixels by altering the distribution of the pixels and obliterating the statistical features of the image. Figure 10 shows the histograms of the original and encrypted images of the Lena, Fruits, Tree, and Peppers images. As can be seen from the figure, the frequency distribution of the pixel values of the ciphertext image after encryption is very different from that of the plaintext image. Additionally, the pixel values in the original plaintext image are unevenly distributed, while those in the ciphertext image appear with equal probability. These observations suggest that the encryption process can alter the distribution of the pixel information in the plaintext image, enabling it to be obscured within the ciphertext image to a greater extent.



(**a**) Original Lena (**b**) Original Fruits (**c**) Original Tree (**d**) Original Peppers

(**e**) Encrypted Lena (**f**) Encrypted Fruits (**g**) Encrypted Tree (**h**) Encrypted Peppers

Figure 10. Histogram Analysis Test. (**a**–**d**) are the histograms of the original images; (**e**–**h**) are the histograms of the encrypted images, Colors Red, Green and Blue represent the R, G and B components of the image respectively.

Furthermore, we quantitatively assessed the homogeneity of the histograms using the variance and chi-square (χ^2), respectively. The variance of the histogram is calculated as:

$$Var(C) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(c_i - c_j)^2, \tag{8}$$

where c is the vector of histogram values, c_i and c_j are the gray values i and j, respectively. Table 7 shows the variance of the plaintext and ciphertext images of the test images. According to the experimental data, it can be seen that the variance value of the image

changes drastically before and after encryption; the lower the variance value, the better the consistency of the image.

Table 7. Histogram variance test.

Image	Original Image			Encrypted Image		
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**
Lena	71,807.81	35,955.32	94,033.58	246.64	225.65	276.99
Fruits	500,913.19	711,257.72	1,691,169.31	1087.46	984.19	997.47
Tree	81,690.41	57,232.69	130,333.24	336.59	264.72	253.50
Peppers	856,092.98	1,278,525.96	1,973,421.39	849.86	1055.53	794.71

The chi-square (χ^2) of the histogram is calculated by

$$\chi^2 = \sum_{i=0}^{255} \frac{(C_i - FR)^2}{FR}, \tag{9}$$

where $FR = (M \times N)/256$ is the expected frequency of each gray value, and c_i is the gray value i. The chi-square (χ^2) values of the histogram for the test images are shown in Table 8. The data in the table are less than the critical value of A, which means that the histogram of the encrypted images is uniform.

Table 8. Histogram chi-square (χ^2) test.

Image	Encrypted Image χ^2			χ^2	
	Red	**Green**	**Blue**	**1%**	**5%**
Lena	245.68	224.77	275.91	310.46	293.25
Fruits	288.86	261.43	264.95		
Tree	335.27	263.69	252.51		
Peppers	221.63	262.85	197.90		

Therefore, the encryption algorithm proposed in this paper is better able to resist histogram attacks.

5.3. Correlation Analysis

Digital images typically have a high degree of data redundancy, and there can be extremely high correlations between adjacent pixels. Image-encryption algorithms should effectively break the correlation between pixels to minimize their correlation and improve the security of image encryption. The formula for calculating the correlation coefficient is defined as:

$$C_{mn} = \frac{\sum_{i=1}^{N} \left(m_i - \frac{1}{N}\sum_{i=1}^{N} m_i\right)\left(n_i - \frac{1}{N}\sum_{i=1}^{N} n_i\right),}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(m_i - \frac{1}{N}\sum_{i=1}^{N} m_i\right)^2} \times \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(n_i - \frac{1}{N}\sum_{i=1}^{N} n_i\right)^2}}, \tag{10}$$

where m_i and n_i are the neighboring pixel values, and N is the number of selected pixel pairs. In this experiment, test images were computed for 6000 pairs of randomly selected adjacent points in the horizontal, vertical, and diagonal directions. Figure 11 shows the neighboring pixel distribution of the original and encrypted images of the Lena image. The correlation data for the test images are given in Table 9. The results show that the correlation coefficients of the plaintext images of the four types of images are very high, indicating that the pixel values of the adjacent pixels are very different and lack pixel independence, while the correlation coefficients of the encrypted ciphertext images are almost close to zero, proving that the scrambling method and the diffusion method of the images are very good and the correlation between the pixels can be effectively broken.

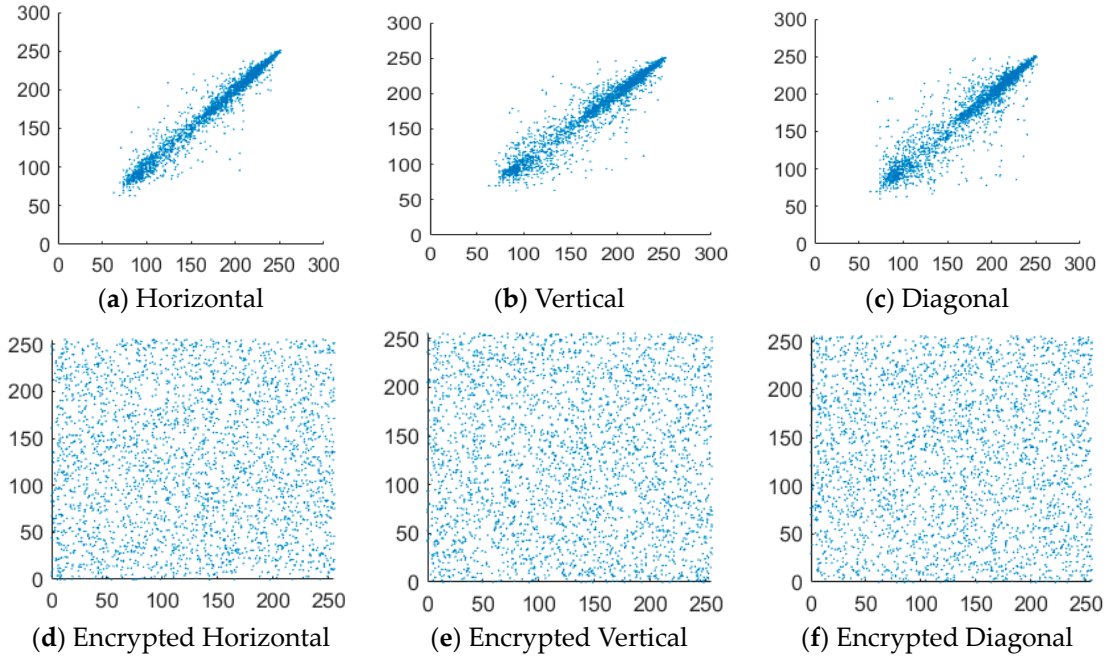


Figure 11. Distribution of adjacent pixels. (**a**–**c**) are the horizontal, vertical, and diagonal of Lena; (**d**–**f**) are horizontal, vertical, and diagonal of encrypted Lena.

Table 9. The results of correlation analysis.

Image		Original Image			Encrypted Image		
		Horizontal	**Vertical**	**Diagonal**	**Horizontal**	**Vertical**	**Diagonal**
Lena	R	0.95091	0.97427	0.92734	0.034691	−0.0057101	0.0058595
	G	0.93313	0.96627	0.91133	−0.005392	−0.006838	−0.0010488
	B	0.90297	0.93938	0.87786	0.042085	0.014217	0.0067408
Fruits	R	0.99213	0.99218	0.9853	−0.011436	0.0091462	−0.014984
	G	0.98765	0.98552	0.97358	0.005756	0.0005085	−0.014769
	B	0.95337	0.92875	0.89063	−0.028683	−0.023687	−0.007692
Tree	R	0.94871	0.95743	0.91769	0.0076669	−0.017613	0.0048572
	G	0.93514	0.94017	0.88554	−0.0069922	0.0034771	−0.0023768
	B	0.96856	0.96686	0.94028	0.0031961	0.0024717	−0.020046
Peppers	R	0.9589	0.96447	0.9527	0.0081944	0.0017356	0.0061784
	G	0.98078	0.98024	0.96711	−0.0083731	−0.0084463	0.014309
	B	0.9652	0.96509	0.94383	−0.014572	0.0039651	0.019823
4.1.01	R	0.97286	0.96355	0.94836	0.028503	0.0050138	−0.014723
	G	0.97139	0.96366	0.9462	0.0007874	0.016867	−0.006223
	B	0.95672	0.95208	0.9345	0.00027603	−0.025448	0.0068728
4.1.03	R	0.97769	0.92788	0.91315	−0.029438	−0.000063	0.005256
	G	0.97624	0.91268	0.89888	−0.02618	−0.010079	0.0047882
	B	0.97391	0.91506	0.9007	0.0019779	0.015966	−0.01633
4.1.05	R	0.96806	0.93586	0.91441	0.010278	−0.016679	−0.012654
	G	0.98081	0.94518	0.92949	−0.011318	0.010572	0.0051661
	B	0.98181	0.97654	0.96298	0.0072627	0.012419	0.0017892
4.2.01	R	0.99382	0.99484	0.9895	0.0006253	0.008218	0.01858
	G	0.98232	0.98963	0.97436	0.0046303	0.010198	0.014741
	B	0.98499	0.98034	0.96777	−0.008814	−0.023014	−0.020686
4.2.03	R	0.92155	0.86798	0.8538	0.0003608	0.000302	0.0022013
	G	0.86477	0.77449	0.74542	−0.0057717	0.012188	−0.01448
	B	0.90892	0.88541	0.84698	0.013629	0.018246	0.002772
4.2.05	R	0.97152	0.95631	0.93385	−0.0082287	0.004509	−0.006422
	G	0.95994	0.96712	0.93407	−0.0072216	−0.012066	0.0076239
	B	0.96319	0.93949	0.91802	−0.0020481	−0.034602	−0.0043855

5.4. Information Entropy Analysis

Information entropy is a crucial metric for measuring the randomness of encryption. It is a physical quantity that reflects the level of confusion in the pixels of a ciphertext image. The degree of order within a system is inversely proportional to its information entropy. The more ordered the system, the lower the information entropy; conversely, the higher the information entropy, the more chaotic the system. Information entropy is a measure used to quantify the level of organization in a complex system and is defined as follows:

$$H(x) = \sum_{i=0}^{255} P(x_i) \log_2 \frac{1}{P(x_i)}, \tag{11}$$

where x_i is the pixel value, $P(x_i)$ is the probability of a pixel occurring. If the information entropy of an encrypted image is closer to 8, the arrangement of pixels in the ciphertext image is more random, and therefore, the encryption effect is better. By comparing the data in Table 10, the information entropy of the encryption algorithm designed in this paper is closer to 8. Furthermore, the average result of information entropy result for the Lena image is superior to that of other algorithms [2,8,13,26]. Additionally, it can be observed that the color images encrypted using the algorithm proposed in this paper exhibit a high level of randomness.

Table 10. Information entropy test.

Image	Red	Blue	Green	Average
Lena	7.9974	7.9972	7.9974	7.9973
Fruits	7.9992	7.9992	7.9992	7.9992
Tree	7.9973	7.9976	7.9974	7.9974
Peppers	7.9993	7.9992	7.9992	7.9992
4.1.01	7.9971	7.9967	7.9972	7.9970
4.1.03	7.9969	7.997	7.9972	7.9970
4.1.05	7.9973	7.9973	7.9972	7.9973
4.2.01	7.9993	7.9994	7.9994	7.9994
4.2.03	7.9993	7.9993	7.9993	7.9993
4.2.05	7.9993	7.9994	7.9992	7.9993
Lena in Ref. [2]	7.9565	7.9880	7.9828	7.9758
Lena in Ref. [8]	7.9912	7.9914	7.9915	7.9914
Lena in Ref. [13]	7.9949	7.9945	7.9941	7.9945
Lena in Ref. [26]	7.9973	7.9965	7.9969	7.9969

5.5. Differential Attack Analysis

A differential attack is a type of selective plaintext attack that looks for a link between plaintext and ciphertext by tracking the effect of small changes in the plaintext on the ciphertext and using the established link to recover the ciphertext without a key. To resist differential analysis, encryption algorithms are highly sensitive to the plaintext image, and small changes in the plaintext image encrypted by the encryption algorithm will produce a completely different ciphertext image. The algorithm's resistance to differential attacks is typically assessed using two key measures: NPCR is the ratio of pixels with different pixel values at the corresponding positions of two images to the total number of pixels, which has a theoretical value of 99.6094%, and UACI is the average of the ratios of the difference of the pixels at the corresponding positions of the two images to the ratio of 255. UACI is the average value of the ratio of the difference of all pixels at corresponding positions between two images to 255, which calculates the extent to which the pixels at the corresponding positions of two images are not the same, and its ideal value is 33.4635%. Their formulae are as follows:

$$NPCR(C_1, C_2) = \frac{1}{M \times N} \sum_i^M \sum_j^N |Sign(C_1(i,j) - C_2(i,j))| \times 100\%, \tag{12}$$

$$UACI(C_1, C_2) = \frac{1}{M \times N} \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\%, \quad (13)$$

where M and N represent the size of the image, C_1 and C_2 are two compressed and encrypted images that differ from the original image by only one pixel. As shown in Table 11, the test results closely match the ideal values. This indicates that the ciphertext image obtained from two plaintext images with a difference of only one pixel is significantly varied. Therefore, the algorithm proposed in this paper is sensitive to plaintext changes and provides better resistance against differential attacks.

Table 11. NPCR and UACI evaluation.

Image	NPCR (%)			UACI (%)		
	Red	**Green**	**Blue**	**Red**	**Green**	**Blue**
Lena	99.6078	99.5926	99.6063	33.4571	33.4529	33.6494
Fruits	99.5959	99.62	99.6236	33.4779	33.4249	33.4115
Tree	99.646	99.6185	99.6048	33.1465	33.3823	33.389
Peppers	99.6006	99.6082	99.6021	33.5401	33.4724	33.3762
4.1.01	99.617	99.5956	99.5941	33.5234	33.2569	33.343
4.1.03	99.6368	99.5987	99.6399	33.5643	33.3784	33.5677
4.1.05	99.6094	99.5895	99.6155	33.5724	33.4664	33.3415
4.2.01	99.6086	99.6006	99.6109	33.406	33.4824	33.466
4.2.03	99.6063	99.6086	99.6086	33.4551	33.4352	33.4769
4.2.05	99.6052	99.5956	99.6143	33.4651	33.534	33.3952

5.6. Robustness Analysis

Ciphertext images can be contaminated by different types of noise during transmission, making it difficult to correctly recover the original image. Typically, noise attacks and occlusion attacks are used to test the robustness of the system. The anti-noise capability of the encryption algorithm can be assessed by utilizing the Peak Signal-to-Noise Ratio (PSNR). PSNR is calculated mathematically using the following equations:

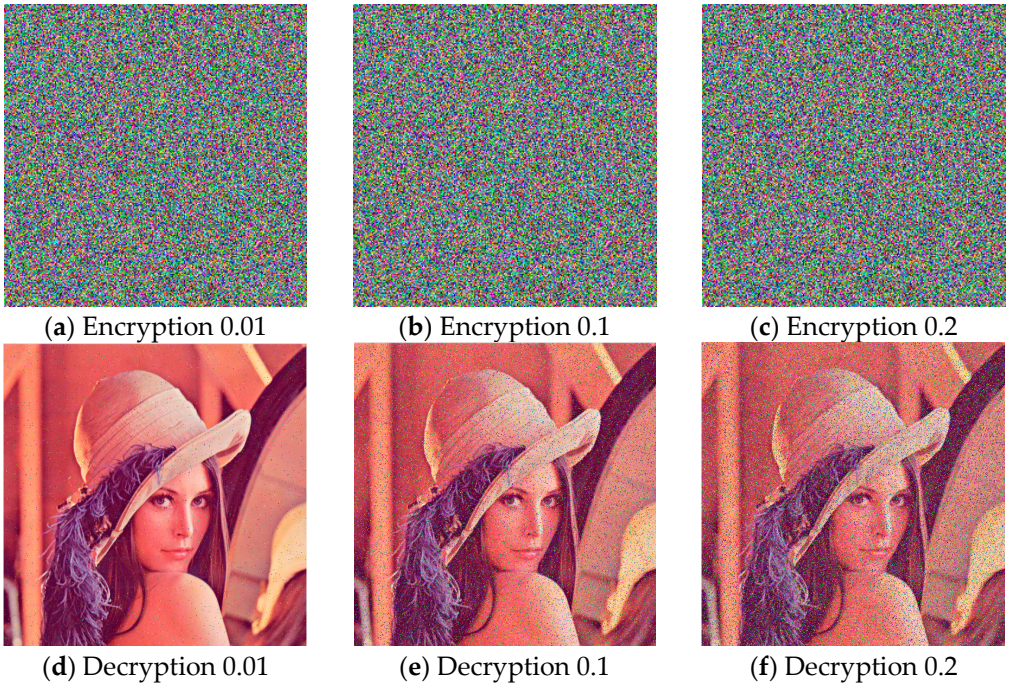
$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (C_1(i,j) - C_2(i,j))^2, \quad (14)$$

$$PSNR = 10 \times lg\left(\frac{255^2}{MSE}\right), \quad (15)$$

where M and N refer to the height and width of the image, $C_1(i,j)$ and $C_2(i,j)$ are the respective pixels located at (i,j) in the original image and the ciphertext image. Taking the color Lena image (512×512) as an example, the robust performance of the proposed algorithm is analyzed.

5.6.1. Noise Attack Analysis

In this subsection, the noise attacks are tested using a Salt and Pepper Noise (SPN) attack. First, SPN with different densities are added to the Lena ciphertext image. Then, the proposed encryption algorithm is applied for decryption. Figure 12a–c shows the images after the SPN attack with noise densities of 0.01, 0.1, and 0.2, while Figure 12d–f displays the corresponding decrypted images. As shown in the figure, with the addition of varying densities of SPN to the encrypted image, most of the information in the original image can be decrypted, although the decrypted image is affected to different degrees. Table 12 lists the PSNR comparison results of the decrypted images with other color encryption algorithm, which show that the proposed algorithm in this paper is noise-resistant.



(**a**) Encryption 0.01 (**b**) Encryption 0.1 (**c**) Encryption 0.2

(**d**) Decryption 0.01 (**e**) Decryption 0.1 (**f**) Decryption 0.2

Figure 12. Decrypted image of Lena after adding different noise SPN attacks. (**a**) SPN 0.01; (**b**) SPN 0.1; (**c**) SPN 0.2; (**d**–**f**) are the corresponding decryption images.

Table 12. PSNR comparison after SPN attacks.

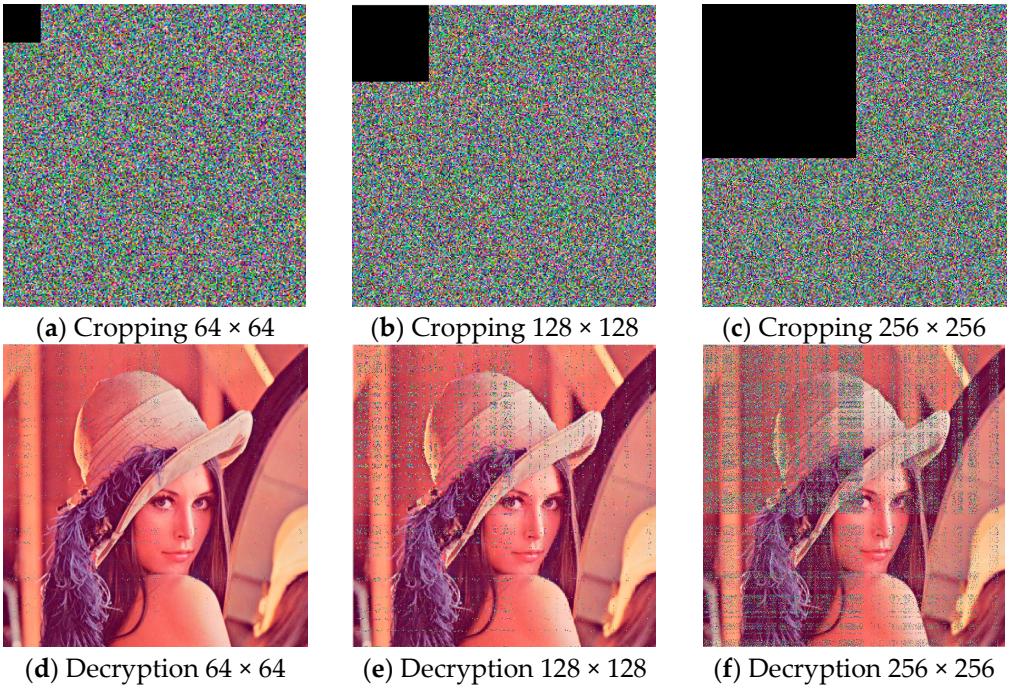
Attacks	Noise Intensity	Proposed			Ref. [11]		
		Red	Green	Blue	Red	Green	Blue
SPN	0.01	32.6758	31.9424	32.8227	28.1835	27.8353	27.9265
	0.1	22.7248	22.0869	22.8662	18.1019	17.8744	17.9727
	0.2	19.6647	19.1715	20.0988	-	-	-

5.6.2. Occlusion Attack Analysis

A cropping attack is a type of covert attack. When a ciphertext image is targeted by a cropping attack, it is necessary to retain as much detailed information as possible to reduce the effect of cropping on the whole image. Figure 13 depicts the Lena ciphertext image being cropped to 64×64, 128×128, and 256×256, along with the corresponding decrypted images. As can be seen from the resultant graph, the decrypted image is slightly blurred, yet the visual content remains discernible, indicating that the algorithm can avoid the impact of some data loss and restore the original image significantly. Table 13 exhibits the PSNR values of the original and decrypted images and the comparison with the existing color encryption algorithm. Therefore, the proposed encryption technique in this paper shows high resilience against data loss attacks.

Table 13. PSNR comparison after cropping attacks.

Block Size	Proposed			Ref. [11]		
	Red	Green	Blue	Red	Green	Blue
64×64	30.6714	32.0929	32.5968	26.1764	25.8628	25.9477
128×128	24.7154	25.7805	26.5331	20.1912	19.8101	20.0184
256×256	18.7519	19.6379	20.5043	14.1626	13.8269	13.9961



(**a**) Cropping 64 × 64 (**b**) Cropping 128 × 128 (**c**) Cropping 256 × 256

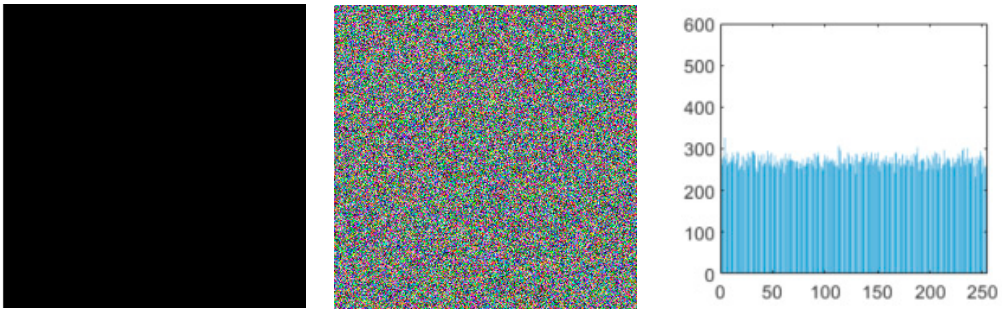
(**d**) Decryption 64 × 64 (**e**) Decryption 128 × 128 (**f**) Decryption 256 × 256

Figure 13. Cropping attack on Lena image with different cropping block sizes. (**a**) 64 × 64; (**b**) 128 × 128; (**c**) 256 × 256; (**d**–**f**) are the corresponding decryption images.

5.7. Classical Types of Attacks Analysis

This section demonstrates the ability of encryption systems to withstand classic kinds of attacks, including chosen plaintext attacks, chosen ciphertext attacks, ciphertext-only attacks, and known plaintext attacks. Objective evidence is crucial, as attackers typically select all-black or all-white original images to expose vulnerabilities in the cryptosystem. Figure 14 shows the corresponding encrypted images and histograms for both the all-white and all-black images, each sized 256 × 256. Experimental results show that the encryption method can resist chosen plaintext attacks and known plaintext attacks.

In addition, it is worth noting that FLCS is highly sensitive to initial values. In the encryption algorithm designed in this paper, a portion of the initial values of FLCS is generated using the original image. This generates chaotic sequences that are highly sensitive to the original image and are subsequently used for the Zig-Zag transform and extended DNA coding encoding. Therefore, during the process of image scrambling, each pixel is associated with other pixels, and small changes in the pixel values can produce an avalanche effect. The proposed algorithm can withstand ciphertext-only attacks and chosen ciphertext attacks.



(**a**) Black image (**b**) Black ciphertext image (**c**) Black ciphertext Histogram

Figure 14. *Cont.*

(**d**) White image (**e**) White ciphertext image (**f**) White ciphertext Histogram

Figure 14. Plaintext attack test. (**a**) Black image; (**b**) Ciphertext image of black image; (**c**) Histogram of ciphertext image of black image; (**d**) White image; (**e**) Ciphertext image of white image; (**f**) Histogram of ciphertext image of white image.

5.8. Time Complexity

In the proposed encryption algorithm, the most time-consuming procedures include generating chaotic sequences via FLCS, performing Zig-Zag transformation, and conducting extended DNA computation. To encrypt an image with size $M \times N$, the initial step involves attaining the chaotic sequences x_1, x_2, x_3, x_4, with a time complexity of $\Theta(M \times N)$. The time complexity for the Zig-Zag transformation is also $\Theta(M \times N)$. Meanwhile, the time complexity of extended DNA computation equates to $\Theta(3 \times M \times N)$. Hence, the final time complexity of the proposed algorithm is $\Theta(3 \times M \times N)$. The encryption and decryption time for a color Lena image of size 256×256 is 1.911 s and 1.946 s, respectively, under this computer configuration.

6. Conclusions

In this paper, the complex dynamic characteristics of FLCS were analyzed, demonstrating its chaotic phenomena and consequent generation of chaotic sequences. The traditional Zig-Zag transform method and DNA coding rule were then extended, resulting in the creation of the block Zig-Zag transform method and 8-bit DNA coding rule for image scrambling. To enhance the security of the algorithm, the two methods were combined with the chaotic sequences produced by FLCS to create a new color image encryption algorithm. Finally, the results of experiments and security analyses indicate that the algorithm has a large secret key space and strong sensitivity to secret keys. The information entropy, NPCR, UACI, and correlation coefficients of the encrypted images are nearly equivalent to their theoretical values. Therefore, the encryption algorithm designed in this paper demonstrates a better ability to resist statistical attacks, differential attacks, and noise attacks, and can effectively safeguard the security of transmitted image information.

Author Contributions: Conceptualization, F.M. and Z.G.; methodology, F.M.; software, F.M. and Z.G.; validation, Z.G.; writing—original draft preparation, F.M. and Z.G.; writing—review and editing, F.M.; visualization, F.M.; supervision, F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Science (Natural Science) Foundation of the Jiangsu Higher Education Institutions of China, grant number 23KJA120004.

Data Availability Statement: Same as many papers in the same research direction, all test images are sourced from the web and publicly available databases. Fruits is available at raw.githubusercontent.com/qiangel1/test-image/main/fruits.bmp, Tree is available at https://sipi.usc.edu/database/database.php?volume=misc&image=6#top, and the Peppers is available at https://sipi.usc.edu/database/database.php?volume=misc&image=13#top.

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

References

1. Telem, A.N.K.; Fotsin, H.B.; Kengne, J. Image encryption algorithm based on dynamic DNA coding operations and 3D chaotic systems. *Multimed. Tools Appl.* **2021**, *80*, 19011–19041. [CrossRef]
2. Khan, M.; Rasheed, A. Permutation-based special linear transforms with application in quantum image encryption algorithm. *Quantum Inf. Process.* **2019**, *18*, 298. [CrossRef]
3. Dash, S.; Padhy, S.; Parija, B.; Rojashree, T.; Patro, K.A.K. A Simple and Fast Medical Image Encryption System Using Chaos-Based Shifting Techniques. *Int. J. Inf. Secur. Priv.* **2022**, *16*, 1–24. [CrossRef]
4. Xie, Z.; Sun, J.; Tang, Y.; Tang, X.; Simpson, O.; Sun, Y. A K-SVD based compressive sensing method for visual chaotic image encryption. *Mathematics* **2023**, *11*, 1658. [CrossRef]
5. Dash, S.; Padhy, S.; Devi, S.A.; Sachi, S.; Patro, K.A.K. An efficient Intra-Inter pixel encryption scheme to secure healthcare images for an IoT environment. *Expert Syst. Appl.* **2023**, *231*, 120622. [CrossRef]
6. Ma, X.; Wang, C.; Qiu, W.; Yu, F. A fast hyperchaotic image encryption scheme. *Int. J. Bifurc. Chaos* **2023**, *33*, 2350061. [CrossRef]
7. Qu, G.; Meng, X.; Yin, Y.; Wu, H.; He, W. Optical color image encryption based on Hadamard single-pixel imaging and Arnold transform. *Opt. Lasers Eng.* **2021**, *137*, 106392. [CrossRef]
8. Kumar, M.; Sathish, G.; Alphonse, M.; Lahcen, R.A.M. A new RGB image encryption using generalized heat equation associated with generalized Vigen'ere-type table over symmetric group. *Multimed. Tools Appl.* **2019**, *78*, 28025–28061. [CrossRef]
9. Patro, K.A.K.; Kumar, M.P.; Acharya, B. An efficient dual-stage pixel-diffusion based multimedia-image encryption using one-type 1D chaotic maps. *Sādhanā* **2022**, *47*, 161. [CrossRef]
10. Patro, K.A.K.; Acharya, B. An efficient two-level image encryption system using chaotic maps. *Int. J. Inf. Comput. Secur.* **2023**, *21*, 35–69. [CrossRef]
11. Khalid, M.; Sara, T.; Mohamed, M. A novel color image encryption based on fractional shifted Gegenbauer moments and 2D logistic-sine map. *Vis. Comput.* **2023**, *39*, 1027–1044.
12. Gao, X.; Sun, B.; Cao, Y.; Banerjee, S.; Mou, J. A color image encryption algorithm based on hyperchaotic map and DNA mutation. *Chin. Phys. B* **2023**, *32*, 030501. [CrossRef]
13. Arpaci, B.; Kurt, E.; Celik, K. A new algorithm for the colored image encryption via the modified Chua's circuit. *Eng. Sci. Technol.* **2020**, *23*, 595–604. [CrossRef]
14. Yu, F.; Kong, X.; Mokbel, A.A.M.; Yao, W.; Cai, S. Complex dynamics, hardware implementation and image encryption application of multiscroll memeristive Hopfield neural network with a novel local active memeristor. *IEEE Trans. Circuits Syst. II: Express Briefs* **2022**, *70*, 326–330. [CrossRef]
15. Ren, L.; Mou, J.; Banerjee, S.; Zhang, Y. A Hyperchaotic Map with A New Discrete Memristor Model: Design, Dynamical Analysis, Implementation and Application. *Chaos Solitons Fractals* **2023**, *159*, 112133. [CrossRef]
16. Wang, L.; Cao, Y.; Jahanshahi, H.; Wang, Z.; Mou, J. Color image encryption algorithm based on Double layer Josephus scramble and laser chaotic system. *Optics* **2023**, *275*, 170590. [CrossRef]
17. Meng, F.; Zeng, X.; Wang, Z.; Wang, X. Anti-Synchronization of Fractional-Order Chaotic Circuit with Memristor via Periodic Intermittent Control. *Adv. Math. Phys.* **2020**, *1*, 5158489. [CrossRef]
18. Liu, T.; Yan, H.; Banerjee, S.; Mou, J. A fractional-order chaotic system with hidden attractor and self-excited attractor and its DSP implementation. *Chaos Solitons Fractals* **2021**, *145*, 10791. [CrossRef]
19. Meng, F.; Zeng, X.; Wang, Z.; Wang, X. Adaptive synchronization of fractional-order coupled neurons under electromagnetic radiation. *Int. J. Bifurc. Chaos* **2020**, *30*, 2050044. [CrossRef]
20. Li, X.; Mou, J.; Cao, Y.; Banerjee, S. An optical image encryption algorithm based on a fractional-order laser hyperchaotic system. *Int. J. Bifurc. Chaos* **2022**, *32*, 2250035. [CrossRef]
21. Guo, Z.; Sun, P. Improved reverse zigzag transform and DNA diffusion chaotic image encryption method. *Multimed. Tools Appl.* **2022**, *81*, 11301–11323. [CrossRef]
22. Gao, H.; Wang, X. An image encryption algorithm based on dynamic row scrambling and ZigZag transform. *Chaos Solitons Fractals* **2021**, *147*, 110962.
23. 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]
24. Enayatifar, R.; Guimarães, F.G.; Siarry, P. Index-based permutation-diffusion in multiple-image encryption using DNA sequence. *Opt. Lasers Eng.* **2019**, *115*, 131–140. [CrossRef]
25. 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]
26. Rehman, A.; Liao, X. A novel robust dual diffusion/confusion encryption technique for color image based on Chaos, DNA and SHA-2. *Multimed. Tools Appl.* **2019**, *78*, 2105–2133. [CrossRef]
27. Yildirim, M. Optical color image encryption scheme with a novel DNA encoding algorithm based on a chaotic circuit. *Chaos Solitons Fractals* **2022**, *155*, 111631. [CrossRef]
28. Wang, X.; Su, Y. Image encryption based on compressed sensing and DNA encoding. *Signal Process. Image Commun.* **2021**, *12*, 116246. [CrossRef]
29. Yan, X.P.; Wang, X.Y.; Xian, Y.J. Chaotic image encryption algorithm based on arithmetic sequence scrambling model and DNA encoding operation. *Multimed. Tools Appl.* **2021**, *80*, 10949–10983. [CrossRef]

30. Wang, X.; Su, Y.; Liu, L.; Zhang, H.; Di, S. Color image encryption algorithm based on Fisher-Yates scrambling and DNA subsequence operation. *Vis. Comput.* **2023**, *39*, 43–58. [CrossRef]
31. Zhang, Y.B.; Zhang, L.; Zhong, Z.; Yu, L.; Shan, M.; Zhao, Y. Hyperchaotic image encryption using phase-truncated fractional Fourier transform and DNA-level operation. *Opt. Lasers Eng.* **2021**, *143*, 106626. [CrossRef]
32. Li, P.; Xu, J.; Mou, J. Fractional-order 4D hyper-chaotic memristive system and application in color image encryption. *EURASIP J. Image Video Process.* **2019**, *2019*, 22. [CrossRef]
33. Haddad, I.; Herbadji, D.; Belmeguenai, A.; Boumerdassi, S. Colour image encryption based on an improved fractional-order logistic map. *Int. J. Electron. Secur. Digit. Forensics.* **2023**, *15*, 66–87. [CrossRef]
34. Alexan, W.; Alexan, N.; Gabr, M. Multiple-layer image encryption utilizing fractional-order chen hyperchaotic map and cryptographically secure prngs. *Fractal Fract.* **2023**, *7*, 287. [CrossRef]
35. Li, X.; Mou, J.; Xiong, L.; Wang, Z.; Xu, J. Fractional-order double-ring erbium-doped fiber laser chaotic system and its application on image encryption. *Opt. Laser Technol.* **2021**, *140*, 107074. [CrossRef]
36. Chen, L.; Yin, H.; Huang, T.; Yuan, L.; Zheng, S.; Yin, L. Chaos in fractional-order discrete neural networks with application to image encryption. *Neural Netw.* **2020**, *125*, 174–184. [CrossRef] [PubMed]
37. Banerjee, S.; Saha, P.; Chowdhury, A.R. Chaotic aspects of lasers with host-induced nonlinearity and its control. *Phys. Lett. A* **2001**, *291*, 103–114. [CrossRef]
38. USC-SIPI Image Database [DB/OL]. Available online: https://sipi.usc.edu/database (accessed on 2 August 2023).