mathematics

MDPI

Article

A New Chaos-Based Image Encryption Algorithm Based on Discrete Fourier Transform and Improved Joseph Traversal

Mingxu Wang *, Xianping Fu *, Xiaopeng Yan and Lin Teng

School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; yanxp@dlmu.edu.cn (X.Y.); tenglin@dlmu.edu.cn (L.T.)
* Correspondence: mxwang@dlmu.edu.cn (M.W.); fxp@dlmu.edu.cn (X.F.)

Abstract: To further enhance the security of image encryption, a new chaos-based image encryption algorithm (IEA) based on discrete Fourier transform and Joseph traversal is proposed to encrypt the plain image in both the frequency domain and space domain simultaneously. In the proposed IEA, the logistic map is used to generate the appropriate chaotic sequence, and the improved Joseph traversal is used to scramble the image in different starting positions and variable step sizes. Then, block diffusion is performed at the end. The main finding concerning the proposed IEA is that the combination of discrete Fourier transform and Joseph traversal can enhance the security of the image information, which has been validated by measuring the performance in resisting the common types of attacks.

Keywords: image encryption; Joseph traversal; Fourier transform; logistic map

MSC: 37N99

Citation: Wang, M.; Fu, X.; Yan, X.; Teng, L. A New Chaos-Based Image Encryption Algorithm Based on Discrete Fourier Transform and Improved Joseph Traversal. *Mathematics* **2024**, *12*, 638. https://doi.org/10.3390/math12050638

Academic Editors: José Balthazar, Angelo Marcelo Tusset, Átila Madureira Bueno, Diego Colón and Marcus Varanis

Received: 9 January 2024
Revised: 10 February 2024
Accepted: 20 February 2024
Published: 21 February 2024

1. Introduction

With the fast development of computer information technology, the digital image as a kind of important data carrier is widely used and data security is receiving more attention, so the significance of data security research has become great. Image encryption technology has become an important part of information security, and it is used in many fields, such as engineering, industry and medical treatment. Therefore, to ensure the reliability and security of image data transmission and storage, the image encryption algorithm (IEA) has become a research hotspot [1–4]. The traditional IEAs, such as the data encryption standard [5] and RSA [6], have many defects when facing images with large data capacity and high redundancy, such as the relatively simple key, small key space and small parameters to determine the sequence, and low security performance of the IEAs. Therefore, there will be great disadvantages when it is applied to IEAs. Hence, several IEAs are proposed using different theories and technologies, such as deep learning [7], S-box [8], elliptic curve [9], Josephus traversal [10], DNA operation [11], compression sensing [12], quantum cellular Automata [13], fractional Fourier transform (FRFT) [14], and chaos theory [15–24].

Recently, some chaos-based IEAs have been proposed [25–35] due to the cryptographic properties of chaos theory, such as the initial conditions' sensitivity and unpredictability. Zhu et al. [25] constructed a new one-dimensional fractional chaotic map and applied it in an IEA using parallel DNA coding. The new chaotic map has a larger range of chaotic parameters and better chaotic characteristics, whereas the parallel DNA coding overcomes the shortcoming of common DNA coding-based IEAs. To enlarge the small parameter range and key space of the existing hyperchaotic maps, Zhong et al. [26] proposed a novel wide-range discrete hyperchaotic map (3D-SCC) based on the mathematical model of the Sine map, and they further designed a 3D-SCC-based IEA. Among them, the IEAs in the transform domain usually have better anti-interference performance and higher security [27–35]. Shao et al. [27] introduced a trinion discrete cosine transform that can process color images holistically, which can be computed by a combination of single-channel

discrete cosine transform and then developing a robust color IEA jointing with a quantum logistic map and Josephus traversing. To strike a good balance between the security and efficiency of the IEA, Feng et al. [28] introduced an IEA based on plane-level image filtering and discrete logarithmic transformation. With the increase in the image transmission data, single IEAs are of low efficiency to meet the real-life applications. Hence, double-image and multi-image IEAs have designed [29–35]. Kaur et al. [29] applied a fractional Fourier transform for double image encryption and proposed a reality-preserving scheme to overcome the complex outcome of the transform, which becomes a limitation due to the requirement of double memory for storage and transmission besides computational complexity. By combining FRFT with discrete fractional angular transform, a double-image IEA was designed by Qiu et al. [30]. Shao et al. [31] introduced a double IEA mainly based on the symmetry of two-dimensional discrete Fourier transform and equal modulus decomposition. Tong et al. [32] designed a nonlinear multi-image IEA by combining the reality-preserving discrete fractional angular transform with the deoxyribonucleic acid sequence operations. Based on discrete Fourier transforms (DFTs) and logistic-exponent-sine map, Tang et al. [33] investigated an IEA for multiple color images. When encrypting multiple images, compression is a commonly used method to reduce the transmission burden, which may result in a loss of partial image data and reduce the quality of the decryption image. To address this issue, Wang et al. [34] proposed a double-image compression and encryption scheme based on a chaotic map and reality-preserving multi-order discrete fractional Fourier transform. A secure double-color IEA with a nonlinear operation and a plaintext-related joint permutation–diffusion mechanism was designed Xiao et al. [35] based on the quaternion multiple parameter discrete fractional angular transform.

Motivated by the above discussions, an IEA based on DFT and improved Joseph traversal (IEA-DIJT) is designed in this paper. IEA-DIJT can encrypt the plain image from the frequency domain and spatial domain with a high efficiency and low complexity. Through different analyses and experimental comparison, the proposed IEA-DIJT in this work has been validated to have good security and effectiveness.

The structure of this paper is organized as follows. Section 2 introduces some preliminary works. Section 3 provides the detailed IEA-DIJT. In Section 4, the simulation experiments and security analyses are performed. Finally, Section 5 summarizes this paper and discusses the future work.

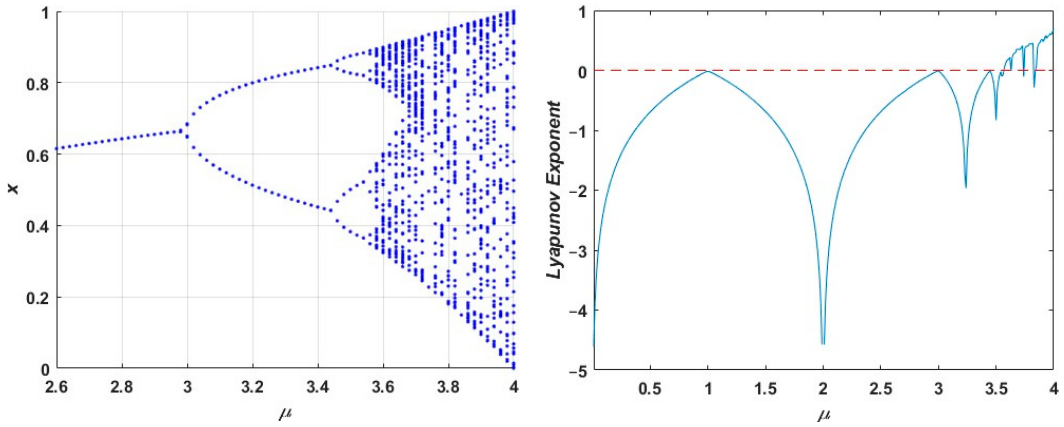
2. Preliminary Works

2.1. Logistic Map

The chaotic map is sensitive to the initial value [10], which means that even if there is a tiny modification in the input parameters, the results obtained will vary greatly. This feature is very suitable for an IEA. The chaotic map used in the IEA-DIJT is the classical logistic map and its formula is:

$$x_{n+1} = \mu x_n(1 - x_n) \tag{1}$$

where $x_{n+1} \in (0, 1)$. μ is the control parameter. Figure 1 respectively depicts the bifurcation diagram and the Lyapunov exponent of Equation (1). As can be seen in Figure 1, Equation (1) performs chaotic behaviors and not periodic when $\mu \in (3.5699456, 4]$.



(**a**) (**b**)

Figure 1. Performance evaluation of the logistic map: (**a**) bifurcation diagram and (**b**) Lyapunov exponent.

2.2. Fourier Transform

Fourier transform, as a classical signal-processing method, has excellent applicability in the field of IEAs. Encrypting images in the transform domain can ensure that the corresponding image information can still be decoded even if some information is lost. Moreover, IEAs based on Fourier transform have the characteristics of high efficiency and high security. Hence, this paper designs an IEA based on Fourier transform. This section details the DFT. For the finite length series $f(x)$, the one-dimensional DFT is:

$$F(u) = DFT[f(x)] = \sum_{x=0}^{N-1} f(x)W^{ux},\ u = 0, 1, \cdots, N-1 \tag{2}$$

and the corresponding inverse transformation is:

$$f(x) = IDFT[F(u)] = \frac{1}{N}\sum_{x=0}^{N-1} F(u)W^{-ux},\ x = 0, 1, \cdots, N-1 \tag{3}$$

where $W = e^{-j\frac{2\pi}{N}}$ is the transformation kernel.

The matrix model of the DFT is:

$$\begin{pmatrix} F(0) \\ F(1) \\ \vdots \\ F(N-1) \end{pmatrix} = \begin{pmatrix} W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^{1\times 1} & W^{2\times 1} & \cdots & W^{(N-1)\times 1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W^0 & W^{1\times (N-1)} & W^{2\times (N-1)} & \cdots & W^{(N-1)\times (N-1)} \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{pmatrix}, \tag{4}$$

and the corresponding inverse transformation is:

$$\begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{pmatrix} = \frac{1}{N}\begin{pmatrix} W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^{-1\times 1} & W^{-2\times 1} & \cdots & W^{-(N-1)\times 1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W^0 & W^{-1\times (N-1)} & W^{-2\times (N-1)} & \cdots & W^{-(N-1)\times (N-1)} \end{pmatrix} \begin{pmatrix} F(0) \\ F(1) \\ \vdots \\ F(N-1) \end{pmatrix} \tag{5}$$

2.3. Improved Joseph Traversal

This section describes the specific progress of the improved Joseph traversal. First, the Joseph traversal is a mathematical problem: encircling n number in a ring, looping through them in order, and deleting the $k-1$th number, and iterating this process from the kth number until the last number is selected. The function of this process is $f(n,k)$. Assuming it starts from 0, it is obvious that the traditional Joseph traversal has a basic

value, which is $f(1,k) = 0$. The first chosen number is $(k-1)\%n$, so the remaining array is $0,\ 1,\ 2,\ 3,\ldots,\ k-3,\ k-2,\ k,\ k+1$. In other words, it needs to start counting from the k again, but the array changes from $0,\ 1,\ 2,\ldots,\ n-1$ to a new array $k,\ k+1,\ k+2,\ldots,\ n-1,\ 0,\ 1,\ 2,\ldots,\ k-2$, which is different from the original starting position, and $f(n,k)$ is no longer applicable to the new array. Hence, a new function $g(n-1,k)$ is established to represent the solution of the Joseph traversal for $n-1$ numbers starting from k. Since the new array is actually a subproblem, the solution of $g(n-1,k)$ and the solution of $f(n,k)$ have the same array number, $g(n-1,k) = f(n,k)$. However, the two functions are different and cannot be recursively expressed, so it needs to establish a mapping that converts $g(n-1,k)$ into the type of $f(n-1,k)$ by converting it between two arrays: $k \to 0$; $k+1 \to 1; \cdots; n-1 \to n-k-1$; $0 \to n-k$; $1 \to n-k+1; \cdots; k-2 \to n-2$. It can be observed that it can use a function $p(x) = (x+n-k)\%n$ to represent the mapping from left to right. Conversely, if a mapping from right to left is required, $q(x) = (x+k)\%n$ can be used. So, the Joseph traversal that operates after mapping is:

$$f(n-1,k) = p(g(n-1,k)) \tag{6}$$

$$g(n-1,k) = p^{-1}(f(n-1,k)) = q(f(n-1),k) = [f(n-1,k)+k]\%n \tag{7}$$

Hence, the recursive formula of the Joseph traversal has obtained:

$$f(n,k) = \begin{cases} 0 & n = 1 \\ [f(n-1,k)+k]\%n & n > 1 \end{cases} \tag{8}$$

Taking the traversal order of 8 numbers as an example, the traditional Joseph traversal is depicted in Figure 2.

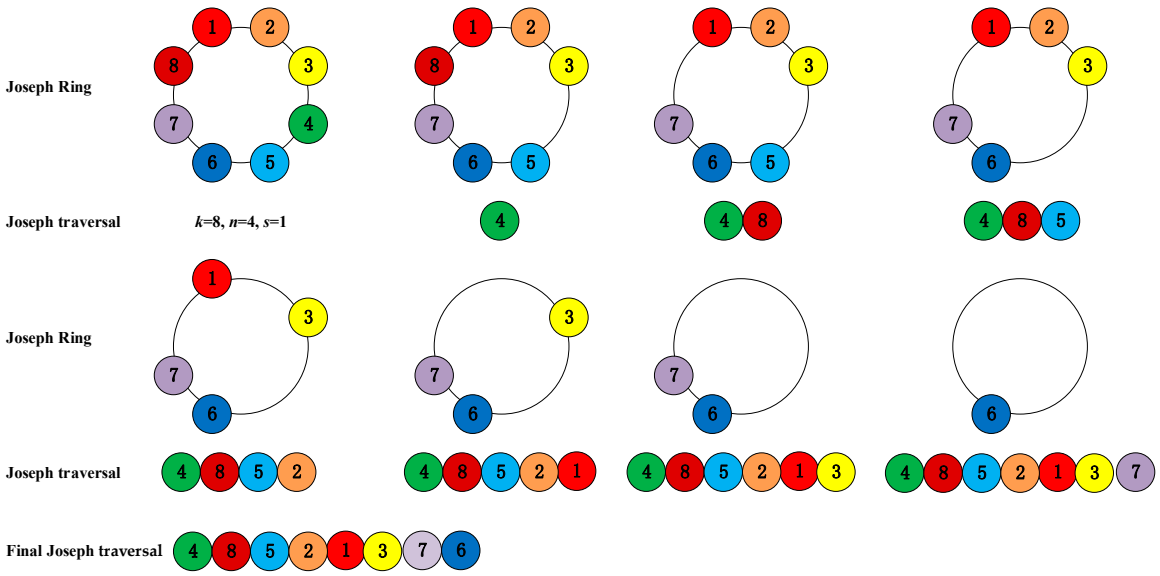


Figure 2. Joseph traversal.

For a good IEA, its encryption process should be flexible and sensitive. For the traditional Joseph mentioned above, it starts from a fixed starting point, traverses with a fixed step size, and finally, obtains a traversal sequence, which is the same for each encryption process. The chosen-plaintext and chosen-ciphertext attacks can easily crack this process, posing a threat to the IEA's security. Therefore, to obtain different and sensitive traversal sequences during each encryption process, it is totally important to improve the fixed starting point and step size of the traditional methods. Hence, this section proposes an improved Joseph traversal. The core idea of this method is that the initialization process of its starting position and traversal step size is highly adaptively associated with the plain

image and key set. Assuming the length of the sequence S to be traversed is $M \times N$, the specific traversal steps are as follows:

Step 1: Iterate a pseudo-random sequence of length $2 \times M \times N$ using Equation (1), and then standardize it via Equation (9):

$$Z(i) = floor[(z(i) \times 10^n) \text{mod} 256] \tag{9}$$

Step 2: Divide Z into two sub-sequences of length $M \times N$, named Z_1 and Z_2, respectively, and perform operations on the first element of Z_1 and Z_2 via:

$$\begin{cases} H_1 = (Z_1(1) + HS_1) \text{mod} M \\ H_2 = (Z_2(1) + HS_2) \text{mod} N \end{cases} \tag{10}$$

where H_1 and H_2 are the calculated starting position. HS_1 and HS_2 represent the hash values based on the plain image.

Step 3: Divide sequence S into two sub-sequences based on the hash value HS_3, named S_1 and S_2, respectively. Perform the improved Joseph traversal on S_1 and S_2 based on the starting position H_1 and H_2. The variable step size and direction of traversal are determined based on the values and their parity. For sequence S_1, it is a traversal process with a variable step size determined by Z_1 and the parity determined by sequence Z_2. Conversely, S_2 is a traversal process with a variable step size determined by Z_2 and the parity determined by Z_1. The meaning of parity here is to traverse counterclockwise if the value is odd, and clockwise if not.

Step 4: Combine the two traversal subsequences into the final sequence.

The traversal results of the Joseph traversal are further expanded as a result of the above improvements. The original Joseph traversal is improved to a Joseph traversal model with the variable step sizes, directions, and starting positions.

2.4. Hash Function

By using hash functions, hash values with specific length and irreversibility can be obtained, which are highly in line with the requirements of image encryption algorithms and are often used to resist common attacks. However, in recent years, some IEAs using SHA-1 have shown weak security. By comparison, SHA-2-based IEAs still exhibit better security. As a hash function in SHA-2, SHA-512 performs tremendous collision resistance, making it difficult to observe two different input messages that produce the same hash value. Even if a slightly change to the input message occurs, the generated hash array is totally different. Moreover, due to the robust avalanche effect of SHA-512, it is widely utilized in the field of IEAs. Therefore, this paper uses the SHA-512 in the IEA-DIJT. It can produce a hexadecimal number with a length of 128, which is used as the main part of the key set for the IEA-DIJT.

3. The Specific Operations of the IEA-DIJT

This section details the whole operations of the IEA-DIJT in this work, which includes two main phases: key generation and encryption progress.

3.1. Key Generation

The key set of the IEA-DIJT is mainly the hash value keys, named HV, which is generated based on the plain image. Then, initialize the parameters of the IEA-DIJT using the obtained HV. The following content provides the specific initialization process:

Step 1: Generate the hash array HV using the SHA-512 function. Read the plain image P into the function to generate a hexadecimal number HS with a length of 128, then divide HS into 6 parts and convert into a decimal as follows:

$$HS = hash(PI, \text{'SHA} - 512') \tag{11}$$

$$HV = hex2dec(HS(i : i + 21)), i = 1, 22, \cdots, 106 \tag{12}$$

Among Equations (11) and (12), the function $hash(PI, \text{'SHA} - 512')$ is used to calculate the SHA-512 hash value of the plain image P, whereas the function $hex2dec(\cdot)$ converts a hexadecimal string into the decimal format. $HS(i : i + 21)$ are the values of HS from the i-th to the $i + 21$-th element.

Step 2: Select 6 hash values from HV as the initial states $x_i(0)(i = 1, 2, 3)$ and control parameters $\mu_i(i = 1, 2, 3)$ of Equation (1) (please see Equation (13)), which are the secret keys. For $\mu_i(i = 1, 2, 3)$, this work limits its value to within $[3.59, 4]$ to ensure that the obtained sequence values have a good chaotic state.

$$\begin{cases} x_1(0) = Key_1 \\ x_2(0) = Key_2 \\ x_3(0) = Key_3 \\ \mu_1 = 3.59 + 0.11Key_4 \\ \mu_2 = 3.69 + 0.11Key_5 \\ \mu_3 = 3.79 + 0.11Key_6 \end{cases}. \tag{13}$$

Step 3: Iterate Equation (1) and transform the generated random sequence A_i into a chaotic matrix of the same size as the plaintext matrix. Then, standardize the matrix A_i via:

$$K_i = \text{mod}(floor(A_i \times 10^{13}),\ 255) \tag{14}$$

where K_i is further reshaped into the key matrix K via:

$$K = reshape(K_i, M, N) \tag{15}$$

where M and N are the height and width of the image matrix.

3.2. Description of the IEA-DIJT

Step 1. For the cipher-image hash operation, the SHA-512 function is used to calculate the standard hash sequence and select the corresponding hash values in the hash sequence to calculate the initial value and control parameters of Equation (1):

$$x = H(i). \tag{16}$$

$$\mu(i) = 3.89 + 0.11 \times H(i). \tag{17}$$

Step 2. Read the initial state and control parameter into Equation (1) to generate the chaotic sequence:

$$A(i) = \mu(i) \times x(i) \times (1 - x(i)). \tag{18}$$

Step 3. The discrete forward Fourier transform of the cipher image. By the Fourier transform, the gray distribution function $f(x, y)$ of the image is transformed into the frequency distribution function $F(u, v)$ of the image and obtains the size of the image frequency domain spectrum $(u, v)\|F$ and phase spectrum $\Phi(u, v)$.

Step 4. According to the random sequence of Equation (1), the starting point and step size are calculated, and the amplitude diagram and phase diagram after the Fourier transform are respectively carried out for the forward and backward Joseph traversal.

Step 5. The inverse Fourier transform of the scrambled amplitude graph and phase graph is used to obtain a round scrambled result.

Step 6. The scrambled image P is partitioned, and the image matrix is divided into matrix blocks of the same size, denoted as $F(I)$, where I is the number of blocks.

Step 7. The chaotic sequence A_1 is used to sort and scramble the matrix, and the sequence A_2 is sorted to obtain the index sequence C_1. The matrix F is transformed into a row of matrix P_1. The matrix F is sorted and scrambled by the index, and the scrambling result P_2 is obtained, and P_2 is restored to the same size as the original matrix.

$$\begin{cases} C_1 = sort(A_1(i)) \\ P_1 = reshape(F, 1, i) \\ P_2 = P_1(C_1(i)) \end{cases} . \tag{19}$$

Step 8. For the scrambled image, block diffusion is carried out. Firstly, the first block is diffused, and the diffusion key matrix is selected and generated from the chaotic sequence A_3.

Step 9. The first diffusion matrix is used as the key matrix and the rest of the parts are circularized.

$$\begin{cases} R(i) = P_3(i) \oplus T(i) \\ R(i+1) = R(i) \oplus P_3(i+1) \\ R(i-1) = R(i) \oplus P_3(i-1) \end{cases} . \tag{20}$$

This is the end of the proposed scheme. Figure 3 depicts the above-mentioned operations.

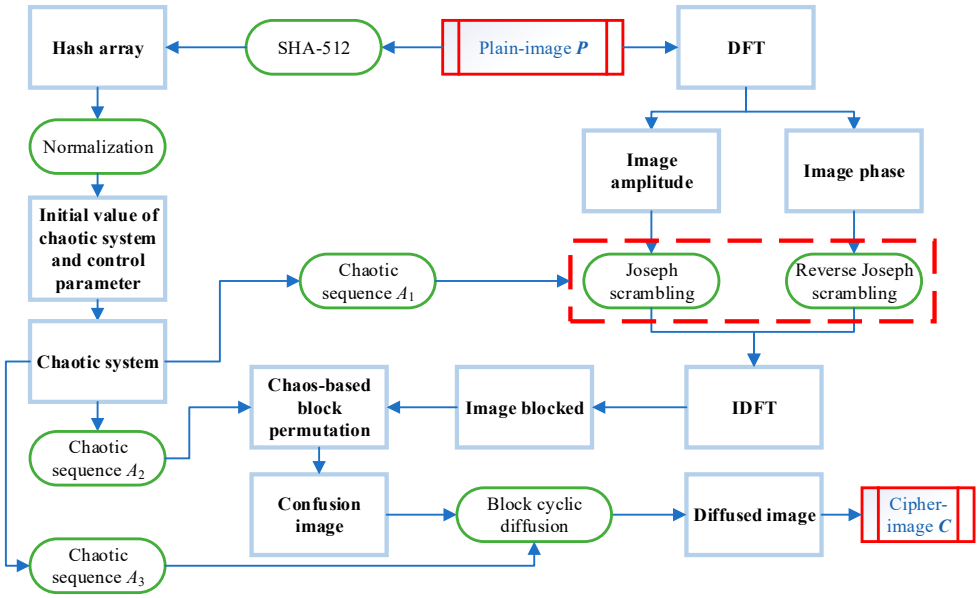


Figure 3. Flow chart of the IEA-DIJT.

4. Experimental Results and Security Analyses

To verify the effectiveness of the IEA-DIJT, this section performs an evaluation of the IEA-DIJT using the digital plain-images. The IEA-DIJT is implemented in the MATLAB R2023b and Windows 10 operating system. Different plain images generate different hash values HV. According to https://ccia.ugr.es/cvg/index2.php and https://sipi.usc.edu/database/database.php (accessed on 3 January 2024), the grayscale plain images were used for the simulation experiments and security analyses.

4.1. Simulation Results

First, Figure 4 plots the cipher images and decoded images of the plain images "Lena" (256 × 256), "Cameraman" (256 × 256), "Elaine" (512 × 512), and "Airport" (1024 × 1024).

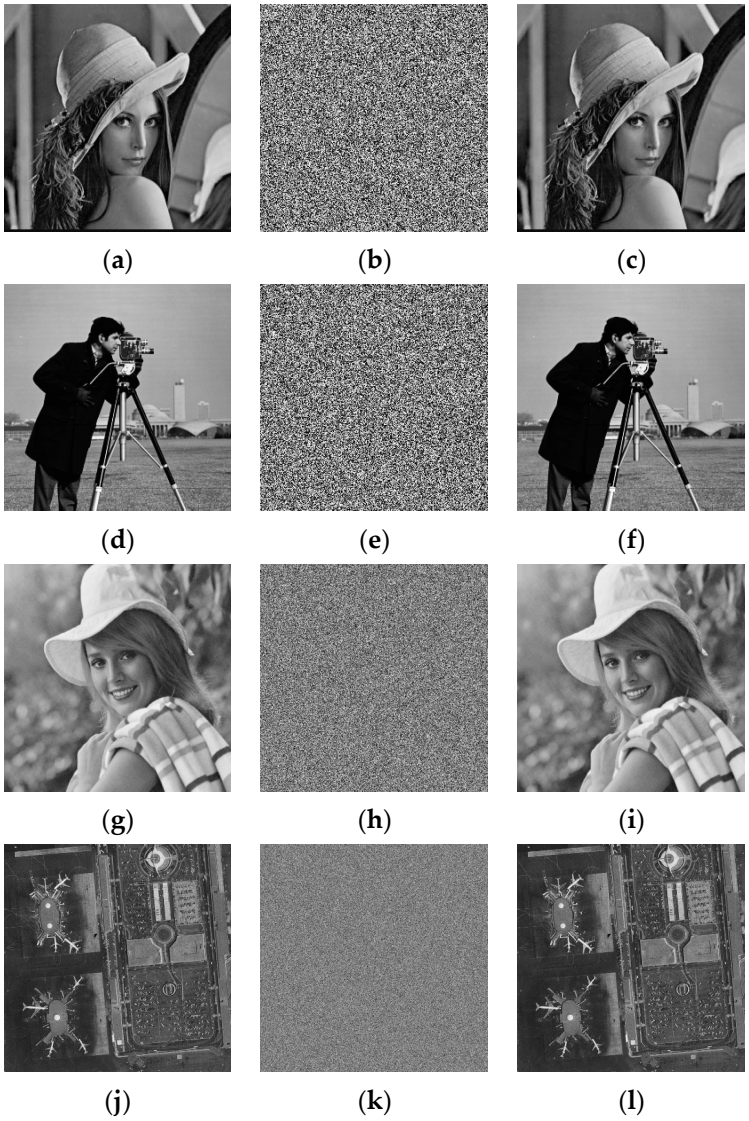


Figure 4. Experimental results: (**a**,**d**,**g**,**j**) are the plain images "Lena", "Cameraman", "Elaine", and "Airport", (**b**,**e**,**h**,**k**) are the corresponding cipher images of (**a**,**d**,**g**,**j**), and (**c**,**f**,**i**,**l**) are the corresponding decrypted images of (**b**,**e**,**h**,**k**).

4.2. Key Sensitivity

If an IEA is highly sensitive to the key stream, it is a good IEA. On the one hand, due to the fact that hash functions are extremely sensitive to the input plain-image message, even a small change can result in a completely different hash value HS. This feature can effectively enhance the key security of the IEA-DIJT. On the other hand, this section further takes the key Key_1 as an example to test the key sensitivity of the IEA-DIJT using the plain image "Lena". The experimental results are shown in Figure 5. When the other keys remain unchanged, Figure 5a is the cipher image generated according to key $x_1(0)$. If the key $x_1(0)$ is changed from 0.8761 to 0.87610000000001, then the obtained cipher image is Figure 5b. Figure 5c is the difference between Figure 5a,b. Figure 5d is the decrypted image generated by decoding Figure 5b. With a minor modification of the secret key, the cipher image generated by the IEA-DIJT is extremely different from the cipher image using the original keys. Moreover, if a tiny change to a pixel is made, the cipher image is as shown in Figure 5e. Figure 5f is the difference between Figure 5a,e. It indicates that the IEA-DIJT is extremely sensitive to the key set.

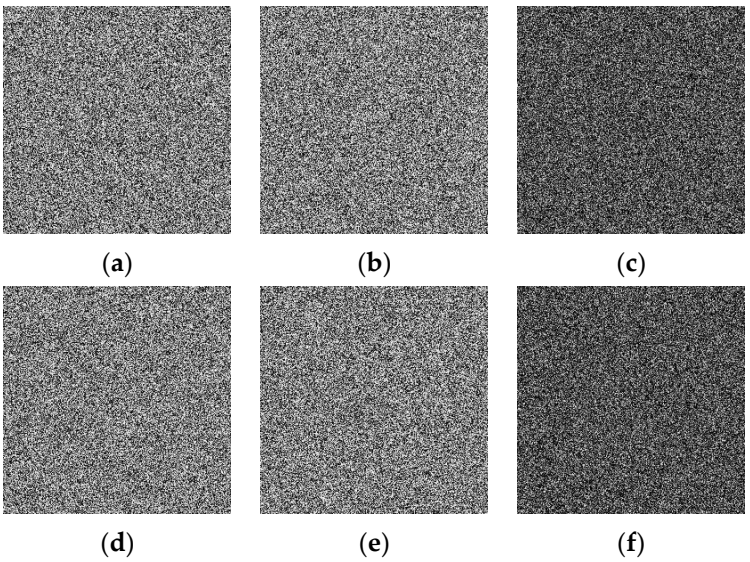


Figure 5. Key sensitivity. The cipher image of the plain image "Lena" using (**a**) $x_1(0)$ and (**b**) $x_1'(0)$, (**c**) the difference between (**a**,**b**,**d**) the decoded image using $x_1'(0)$, (**e**) the cipher image after a tiny change, and (**f**) the difference between (**a**,**e**).

4.3. Histogram

Digital images are composed of pixels, and generally speaking, unsatisfied images have very clear fluctuant properties in the pixel distribution. The pixel values of the cipher image must be roughly distributed to withstand statistical attacks. The histogram of an image provides an accurate and first-hand view of the pixel value spread, and it is often used to find out whether the cipher images obtained by one IEA are robust against statistical attacks. If the histogram of an image looks smooth, the pixel spread is evenly distributed and it is not easy to observe helpful message from the statistical analysis attacks. That is to say, the better the performance of an IEA, the smoother the histogram of the cipher image it obtains.

Figure 6 illustrates the histograms of the plain images "Lena", "Peppers", and "Cameraman" and their corresponding cipher images, respectively. Figure 6a,c,e depict the pixel distribution of the plaint images. For example, there are pixels with a huge number of elements in the image, while some pixels do not even show in the image. In contrast, the pixel distribution in the histogram of the cipher image shown in Figure 6b,d,f is presumably uniform and does not have any distribution features. Therefore, the attackers are unable to exploit statistical attacks to disrupt the IEA-DIJT.

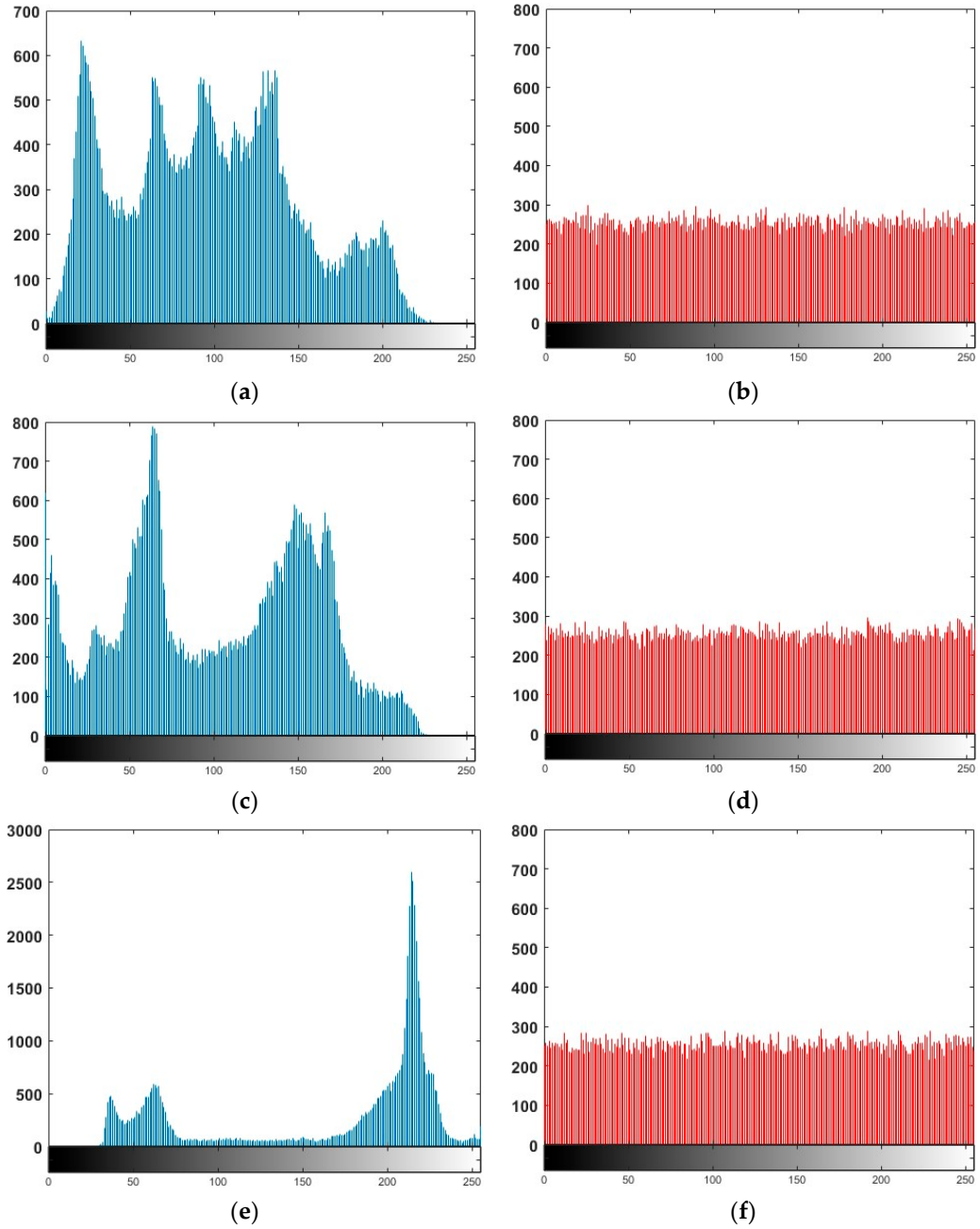


Figure 6. Histograms of the IEA-DIJT: (**a**,**c**,**e**) are the histograms of the plain images "Lena", "Peppers", and "Cameraman", respectively; and (**b**,**d**,**f**) are the histograms of the cipher images "Lena", "Peppers", and "Cameraman", respectively.

4.4. Chi-Square Test

To quantitatively measure the distribution situation of the histogram, this section further uses the chi-square (χ^2) test as a metric. The description of the chi-square [36] is as follows:

$$\chi^2 = \sum_{i=0}^{255} \frac{(q_i - q)^2}{q}, \tag{21}$$

where q_i means the number of pixels with a value of i present in the image, and q is calculated via Equation (22):

$$q = \frac{M \times N}{256}. \tag{22}$$

Table 1 tabulates the χ^2 results of the cipher images, which demonstrate that the obtained χ^2 result was below 293.2483 at a significance level of 5%. Therefore, the IEA-DIJT represents a superior behavior to resist statistical analysis attacks.

Table 1. The obtained χ^2 results.

Size	Image	χ^2 Value	H (0 or 1)	Decision
256	Cameraman	250.6172	0	Accept
	Clock	261.7734	0	Accept
	House	258.7813	0	Accept
	Lena	291.2969	0	Accept
	Moonface	287.2578	0	Accept
	Peppers	233.2734	0	Accept
512	Baboon	283.5293	0	Accept
	Boat	246.8066	0	Accept
	Couple	247.3340	0	Accept
	Elaine	263.0449	0	Accept
	Lena	262.1440	0	Accept
	Peppers	243.7363	0	Accept

4.5. Correlation

The correlation coefficient between two neighboring pixels is used to measure the differences between the plain images and the cipher images. In many plain images, the correlation results between the pixels approach 1. An efficacious IEA should be proposed to break the correlation of the images. This section stochastically picks 10,000 pixels from the plain image "Lena" and the corresponding cipher image, and it estimates the correlations among the pixels in the horizontal (H), vertical (V), and diagonal (D) directions via [37]:

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{23}$$

where

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)),\ D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2,\ E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i. \tag{24}$$

Figure 7 respectively plots the correlations of the plain images and the cipher images. Moreover, Tables 2 and 3 further tabulate the comparison results obtained by different IEAs. The correlation between two neighboring pixels of the plain mages is powerful. In contrast, it is quite weak in its ciphertext. It can be seen from the comparison results that the correlation results of the IEA-DIJT are generally lower compared with the existing IEAs [38–42]. Hence, the IEA-DIJT is effective enough to withstand statistical analysis attacks.

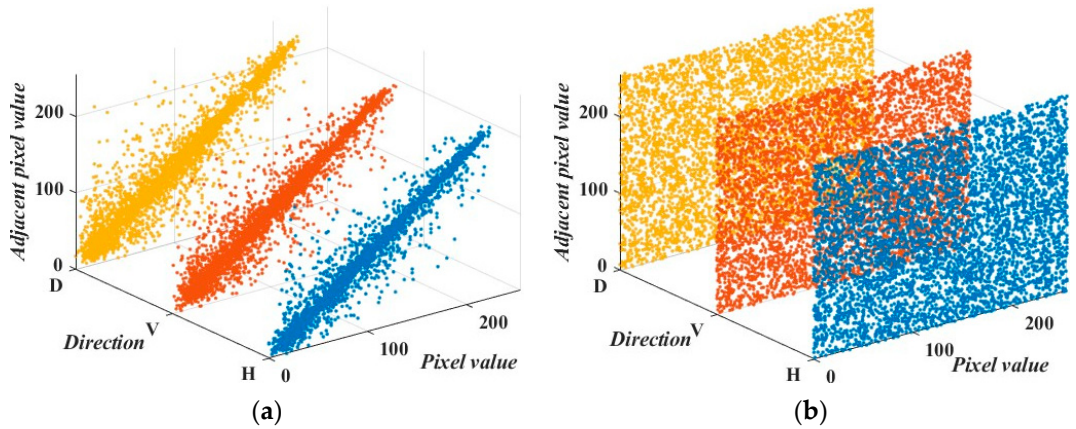


Figure 7. Correlation: the correlation of (**a**) Figure 4a and (**b**) Figure 4b in three directions.

Table 2. Correlation of neighboring pixels of the IEA-DIJT.

Size	Image	Direction	IEA-DIJT		Size	Image	Direction	IEA-DIJT	
			P	*C*				*P*	*C*
256	Cameraman	H	0.8950	−0.0017	512	Baboon	H	0.7267	0.0010
		V	0.8987	0.0022			V	0.7328	−0.0019
		D	0.9311	0.0011			D	0.6838	0.0011
	Clock	H	0.9340	0.000		Lena	H	0.9719	−0.0004
		V	0.9431	−0.0013			V	0.9850	−0.0004
		D	0.9469	0.0007			D	0.9593	0.0002
	Lena	H	0.9143	0.0013		Peppers	H	0.9573	0.0008
		V	0.9039	0.0009			V	0.9667	0.0018
		D	0.9098	0.0007			D	0.9684	0.0007
	Peppers	H	0.9337	−0.0031					
		V	0.9464	−0.0017					
		D	0.9329	−0.0028					

Table 3. Comparison results with other IEAs for the cipher image "Lena".

Schemes	Directions		
	H	V	D
IEA-DIJT	0.0009	0.0007	−0.0031
Ref. [38]	−0.015595	0.028436	−0.007102
Ref. [39]	0.016720	0.006996	0.011698
Ref. [40]	−0.040583	−0.027371	−0.014449
Ref. [41]	0.004555	−0.015623	−0.005112
Ref. [42]	0.009998	0.001372	−0.006567
Ref. [43]	−0.001885	−0.012793	0.007396

4.6. Information Entropy

4.6.1. Global Information Entropy

Information entropy is considered a measure of the degree of order in the motion state of an object. The higher the entropy, the higher the degree of disorder. Its calculation is Equation (25) [37]. Theoretically, the possibility of information leakage decreases gradually as the information entropy approaches 8.

$$H = \sum_{i=0}^{2^n-1} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{25}$$

where $p(m_i)$ means the probability of m_i happening. For image processing, entropy presents the amount of information contained in an image. When an image embodies N grayscale values, if the grayscale values of each pixel are different, the entropy value is the maximum, $H = \log_2 N$, and the image has the biggest amount of message. It can be considered that each pixel is an independent object target with a maximum information content of N. Hence, the bigger the entropy H of the image, the richer the grayscale of the pixels contained in the image, the more uniform the grayscale distribution, the more ground objects in the image, and the greater the message content of the image.

For a good IEA, the information entropies of the cipher images should be close to 8. Tables 4 and 5 list the computed results of the images and comparison results. As can be observed, all the entropy results [38–43] are close to 8. Moreover, the cipher images obtained by the IEA-DIJT are less likely to leak information and have better ability to resist statistical analysis attacks than other IEAs.

Table 4. Information entropy values of the IEA-DIJT.

Size	Image	IEA-DIJT		Size	Image	IEA-DIJT	
		P	*C*			*P*	*C*
256	Cameraman	6.9719	7.9973	512	Baboon	7.3585	7.9992
	Clock	6.7057	7.9972		Boat	7.1914	7.9993
	House	6.4971	7.9971		Couple	7.2010	7.9994
	Lena	7.5534	7.9968		Elaine	7.5060	7.9993
	Moonface	6.7093	7.9968		Lena	7.4474	7.9995
	Peppers	7.5819	7.9974		Peppers	7.5925	7.9993

Table 5. Comparison results of the information entropy.

Size	Image	IEA-DIJT		Ref. [38]	Ref. [39]	Ref. [40]	Ref. [41]	Ref. [42]	Ref. [43]
		P	*C*						
256	Baboon	7.5534	7.9973	7.9302	7.9973	7.9970	7.9973	7.9787	7.9967
	Lena	7.5819	7.9974	7.9144	7.9970	7.9969	7.9973	7.9950	7.9971
512	Baboon	7.3585	7.9992	7.9866	7.9985	7.9992	7.9993	7.9791	7.9993
	Lena	7.4474	7.9995	7.9729	7.9986	7.9992	7.9992	7.9788	7.9993

4.6.2. Local Shannon Entropy

For better analysis, this section further conducts experimental comparisons of the local information entropy (LSE) via Equation (26):

$$H_{k,T_B}(S,L) = \sum_{i=1}^{k} \frac{H(S_{T_B},L)}{k}, \tag{26}$$

where $H(S_{T_B},L)$ denotes the Shannon entropy of non-overlapping image blocks S_{T_B}. It means that each of the $k = 30$ blocks contain $T_B = 1936$ pixels. The image passes the test if the LSE value is located in $[7.901515698, 7.903422936]$.

The IEA-DIJT was evaluated on the plain images. The results, tabulated in Table 6 (Bold values indicate passing the test), demonstrated that the cipher images passed the LSE test with a passing rate of over 90%. This indicates that the IEA-DIJT provides a high degree of randomness and an even distribution of the cipher images.

Table 6. LSE values for different cipher images.

Size	Image	LSE	Size	Image	LSE
256	Cameraman	**7.902814**	512	Baboon	**7.9032934**
	Clock	**7.9018914**		Boat	**7.9020965**
	House	7.9036226		Couple	**7.9013717**
	Lena	**7.9033111**		Elaine	**7.9025729**
	Moonface	**7.9025067**		Lena	**7.902469**
	Peppers	**7.9026073**		Peppers	**7.9019517**
PASS/ALL			11/12		

4.7. Differential Attack

A differential attack can be used to find the vulnerabilities of an IEA. It can help the attackers to break an IEA by recovering the key set or plaintext message from the cipher image. Therefore, a strong IEA must be greatly sensitive to minor changes in the plaintext, which must spread throughout the entire encrypted image. The number of the pixel change rate ($NPCR$) and the unified average change intensity ($UACI$) are common metrics used to

measure the ability of IEAs against those attacks. In this section, they are calculated via Equations (27)–(29) [44]:

$$NPCR = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} D(i,j) \times 100\%, \tag{27}$$

$$UACI = \frac{1}{W \times H} \left(\sum_{i=1}^{W} \sum_{j=1}^{H} \frac{|c_1(i,j) - c_2(i,j)|}{255} \right) \times 100\%, \tag{28}$$

$$D(i,j) = \begin{cases} 0, & c_1(i,j) = c_2(i,j) \\ 1, & c_1(i,j) \neq c_2(i,j) \end{cases}, \tag{29}$$

where $W \times H$ is the total number of pixels in an image. c_1 and c_2 are two encrypted images with a one-bit difference of their original image. The ideal $NPCR$ and $UACI$ scores are near 99.6093% and 33.4635%, respectively.

Moreover, the important values for determining whether the obtained $NPCR$ and $UACI$ have passed the test are in the light of their critical values. For $NPCR$, the critical value N_α^* for different image sizes is calculated via:

$$N_\alpha^* = \frac{L - \Phi^{-1}(\alpha)\sqrt{L/MN}}{L+1}, \tag{30}$$

where α represents the significance level, and $\Phi^{-1}(\alpha)$ denotes the inverse cumulative density function of normal distribution $N(0,1)$. If the obtained is $NPCR$ larger than N_α^*, it is considered that the IEA has passed the test. For $UACI$, there are two strict critical values U_α^{*-} and U_α^{*+} for different image sizes. They are calculated via:

$$\begin{cases} U_\alpha^{*-} = \mu_u - \Phi^{-1}(\alpha/2) \times \sigma_u, \\ U_\alpha^{*+} = \mu_u + \Phi^{-1}(\alpha/2) \times \sigma_u, \end{cases} \tag{31}$$

where

$$\mu_u = \frac{L+2}{3L+3}, \tag{32}$$

$$\sigma_u^2 = \frac{(L+2)(L^2+2L+3)}{18MNL(L+1)^2}, \tag{33}$$

where L denotes the gray level. MN is the total number of pixels in an image.

Keeping the key set unchanged, this section calculates the $NPCR$ and $UACI$ of two encrypted images c_1 and c_2 (please see Table 7) and compares the obtained values with the existing IEAs [38–43] (please see Tables 8 and 9). As we can observe, most of the $NPCR$ and $UACI$ scores have passed the test, proving that the IEA-DIJT has a strong ability against the differential attacks.

Table 7. $NPCR$ and $UACI$ (%) scores of the IEA-DIJT.

Size	Image	*NPCR*	*UACI*	Size	Image	*NPCR*	*UACI*
		$N_{0.05}^* \geq$ **99.5693**	$U_{0.05}^{*-}$**=33.2824** $U_{0.05}^{*+}$**=33.6447**			$N_{0.05}^* \geq$ **99.5893**	$U_{0.05}^{*-}$**=33.3730** $U_{0.05}^{*+}$**=33.5541**
256	Cameraman	99.6096	33.5167	512	Baboon	99.6131	33.5181
	Clock	99.6118	33.4908		Boat	99.6265	33.5226
	House	99.6107	33.5008		Couple	99.6128	33.5269
	Lena	99.6098	33.4754		Elaine	99.6208	33.5054
	Moonface	99.6126	33.4708		Lena	99.6085	33.5172
	Peppers	99.6170	33.5672		Peppers	99.6143	33.5188

Table 8. Comparison results of the *NPCR* (%).

Size	Image	IEA-DIJT	Ref. [38]	Ref. [39]	Ref. [40]	Ref. [41]	Ref. [42]	Ref. [43]
256	Baboon	99.6096	99.1073	99.6216	99.5911	99.6231	99.5941	99.6307
	Lena	99.6098	98.9807	99.6346	99.6323	99.6154	99.6307	99.6506
512	Baboon	99.6131	99.4785	99.6268	99.6627	99.6299	99.5979	99.6147
	Lena	99.6085	99.5033	99.6436	99.6093	99.5845	99.6082	99.6059

Table 9. Comparison results of the *UACI* (%).

Size	Image	IEA-DIJT	Ref. [38]	Ref. [39]	Ref. [40]	Ref. [41]	Ref. [42]	Ref. [43]
256	Baboon	33.4804	33.7694	33.2025	33.5648	33.4941	33.1112	33.1598
	Lena	33.5172	33.0815	33.1984	33.2378	33.3377	33.0881	33.4858
512	Baboon	33.5181	33.7609	33.1173	33.4588	33.4316	33.0019	33.4921
	Lena	33.5172	33.2334	33.6028	33.4873	33.5262	33.0228	33.4209

4.8. Known-Plaintext and Chosen-Plaintext Attacks

The known-plaintext and chosen-plaintext attacks are commonly used security attacks to measure the performance of an IEA. A weak IEA will offer a chance to disclose the relation between the plaintexts and ciphertexts and even deduce the secret keys by choosing arbitrary plaintexts. To avoid this issue, part of key set of the IEA-DIJT is the hash values. It can generate a totally different cipher image when the IEA-DIJT is applied to the same plain image. As shown in Figure 8, the plain image "Lena" is encrypted twice with two different selected hash values and keeps the other keys unchanged. Figure 8a,b are the obtained cipher images, respectively. Figure 8c depicts the difference in each pixel between the two cipher images in Figure 8a,b. Figure 8d,e are the corresponding histograms. Obviously, Figure 8d,e are different. This means that the IEA-DIJT is able to resist the chosen plaintext attacks. Moreover, this section also tests special plain images, such as pure black or white images, with the size 256×256. Figure 9 depicts the obtained cipher images and their corresponding histograms. As can be observed, the IEA-DIJT can withstand the known-plaintext and chosen-plaintext attacks.

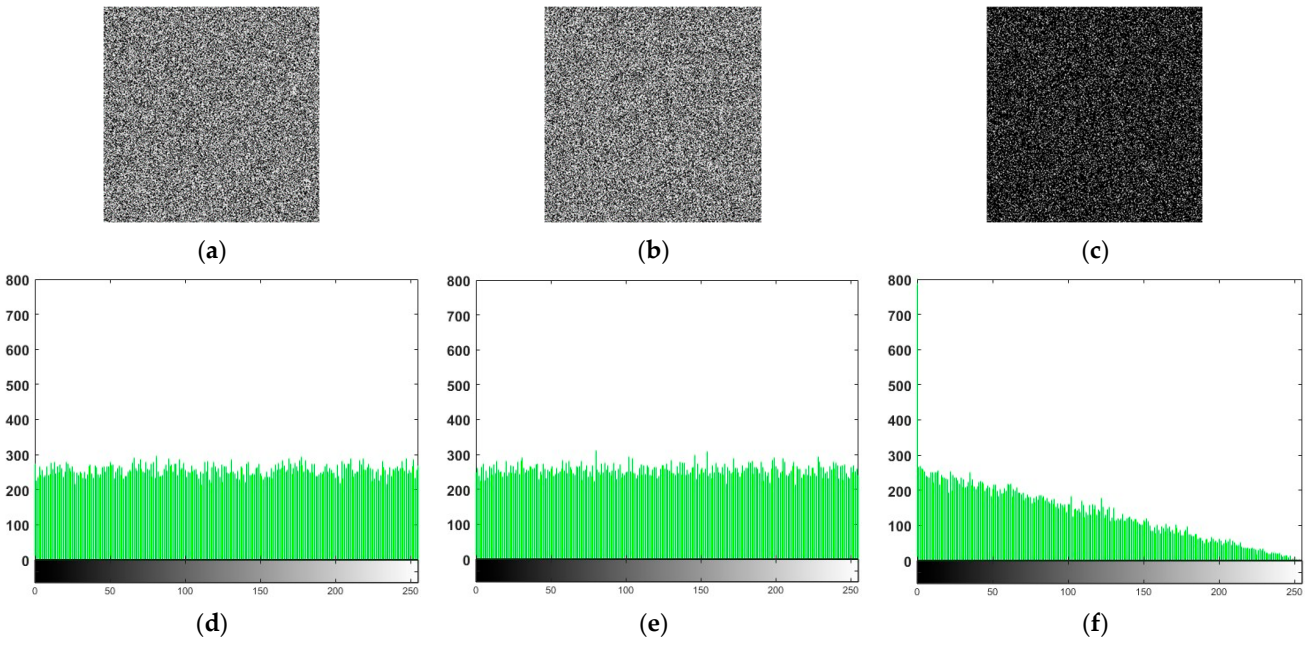


Figure 8. Known-plaintext and chosen-plaintext attacks: (**a**,**b**) are two cipher images with only a one-bit difference in the original image "Lena", and (**c**) the difference between (**a**,**b**,**d**–**f**) are the histograms of (**a**–**c**), respectively.

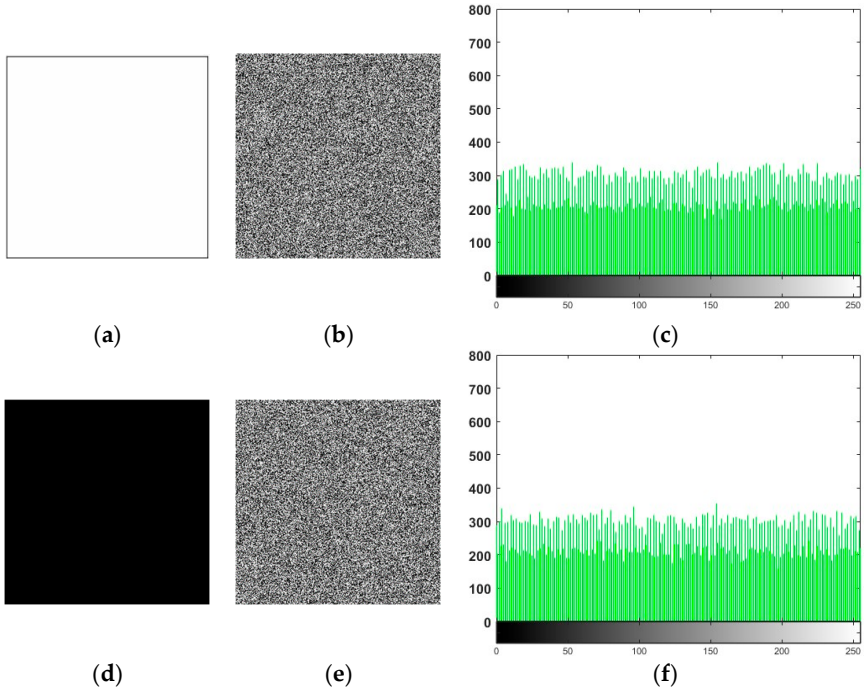


Figure 9. Experimental results of the pure white and black: (**a**) the pure white image, (**b**) the cipher image of (**a**,**c**) the histograms of (**b**,**d**) pure black image, and (**e**) the cipher-image of (**e**,**f**) the histograms of (**e**).

4.9. Robustness to Noise and Data Loss Attacks

Robustness is a significant metric to evaluate the ability of anti-jamming for an IEA. This section tests it via noise and data loss attacks.

4.9.1. Noise Attack

Noise is unavoidably added to the encrypted message in practical communication, which contains Gaussian noise and pepper and salt noise. An efficacious IEA must have the ability to resist these attacks. The salt–pepper noises of different intensities are added to the encrypted image. Figure 10 plots the test results. Figure 10a–c plot the cipher images with the intensity values of 0.01, 0.05, and 0.1, and the corresponding decoded images are depicted in Figure 10d–f. As we can observe, even if the noise intensity is 0.1, the original image can still be recognized. Hence, the IEA-DIJT has a good ability to resist noise attack.

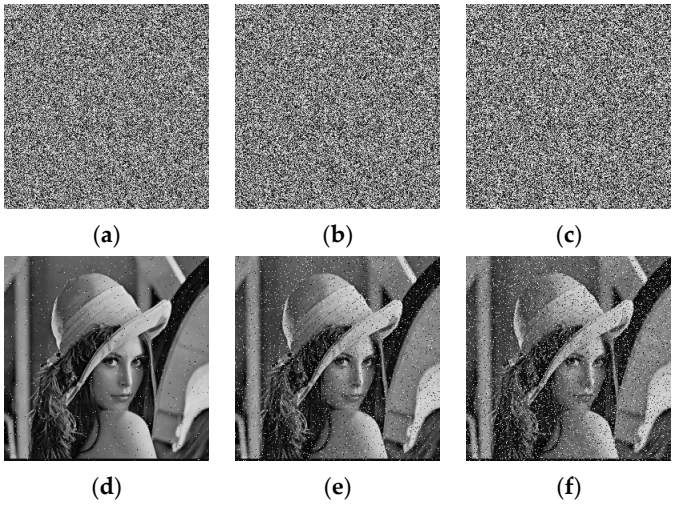


Figure 10. Noise attack. The cipher-image "Lena" with noise intensity (**a**) 0.01, (**b**) 0.05, and (**c**) 0.1, and the decoded images of (**a**–**f**).

4.9.2. Data Loss Attack

As data can be lost during transmission, a data loss attack is necessary to assess how effectiveness the IEA in this paper is. The cipher images are cut off from a certain proportion of data to evaluate the robustness of the IEA, and the reconstruction is then carried out. Clearly, with the increasing proportion of messages, the quality of the decoded images decreases. Figure 11 plots the test results with the data loss proportions 6.08%, 25.32%, 40.20%, and 82.45%. As we can observe, the main information of the original image can be still recognized in the decoded images. Hence, the IEA-DIJT has a good ability to resist data loss attacks. Therefore, the IEA-DIJT is robust.

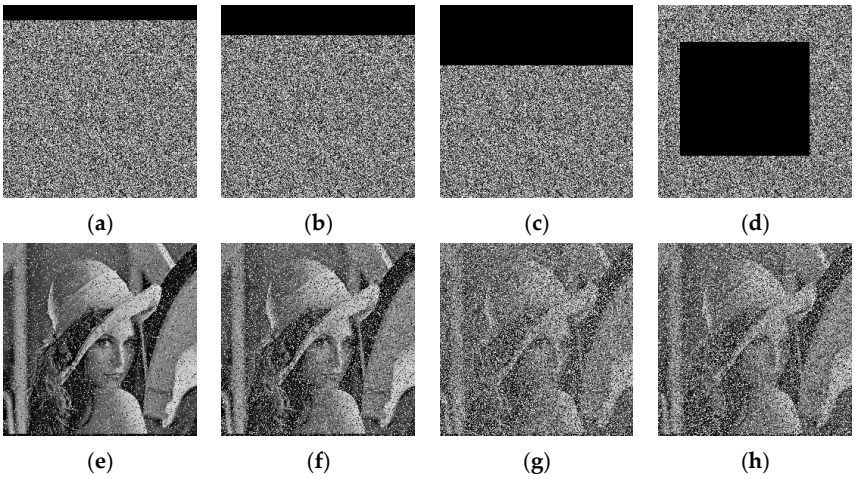


Figure 11. Data loss attack. The cipher-image "Lena" with data loss proportions (**a**) 6.08%, (**b**) 25.32%, (**c**) 40.20%, and (**d**) 82.45%, and the decoded image of (**a**–**h**).

4.10. Complexity and Time

This section evaluates the time and complexity of the IEA-DIJT. Initially, the chaotic sequences are generated by iterating the logistic map. The time taken for this operation is $O(M \times N)$. In the confusion phase, it needs order $O\left(\frac{1}{4} \times M \times N\right)$ times. In the diffusion stage, the time complexity is $O\left(\frac{3}{4} \times M \times N\right)$, so the overall time complexity required to perform the IEA-DIJT is $O(M \times N)$. Moreover, Table 10 tabulates the encryption time of the IEA-DIJT, proving the acceptable speed of the IEA-DIJT.

Table 10. Encryption time (s) results.

Size	Image	Time	Size	Image	Time
256	Cameraman	1.005048	512	Baboon	5.208273
	Clock	0.974152		Boat	5.439504
	House	0.98626		Couple	5.362937
	Lena	1.006378		Elaine	5.394718
	Moonface	0.976231		Lena	5.522115
	Peppers	0.987105		Peppers	5.487128

5. Conclusions

This work proposed a chaos-based IEA called the IEA-DIJT. Initially, the logistic map is used to generate chaotic sequences. Then, the plain image is decomposed into a phase map and an amplitude map using DFT, and different Joseph traversal operations are performed for scrambling. Moreover, the chaos-based diffusion is used to modify the pixel value to make the encryption effect deeper. Finally, an encrypted image is obtained. Through the simulation experiments and theoretical analyses, the security of the IEA-DIJT in resisting the common types of the attacks is verified. Under the premise of ensuring the security of the IEA, the efficiency of the IEA-DIJT is acceptable, which means there is room for improvement in the

efficiency and security. For example, for the chaotic sequence, the IEA-DIJT uses the commonly used logistic map, which can further expand its chaotic region and obtain sequences with better chaotic characteristics. For the DFT-based confusion process, improving it to enhance the efficiency is another worthwhile research direction. Obviously, the improved Joseph traversal and its combination with DFT effectively enhances the security of the IEA-DIJT, increasing the application direction of the classic mathematical Joseph problem and expanding the research tools for IEAs, providing a direction for future research work.

Author Contributions: Methodology, M.W.; Validation, M.W.; Formal analysis, M.W., X.Y. and L.T.; Resources, M.W., X.F. and X.Y.; Writing—original draft, M.W.; Writing—review & editing, M.W. and X.F.; Visualization, X.Y.; Supervision, L.T.; Funding acquisition, X.F. and L.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos: 62176037 and 61701070), the Fundamental Research Funds for the Central Universities (Nos: 3132023252), and the China Postdoctoral Science Foundation (No: 2020M680933).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Daoui, A.; Yamni, M.; Chelloug, S.A.; Wani, M.A.; El-Latif, A.A.A. Efficient Image Encryption Scheme Using Novel 1D Multiparametric Dynamical Tent Map and Parallel Computing. *Mathematics* **2023**, *11*, 1589. [CrossRef]
2. Wang, X.; Zhang, X.; Gao, M.; Tian, Y.; Wang, C.; Iu, H.H.-C. A Color Image Encryption Algorithm Based on Hash Table, Hilbert Curve and Hyper-Chaotic Synchronization. *Mathematics* **2023**, *11*, 567. [CrossRef]
3. Qiu, H.; Zhang, X.; Yue, H.; Liu, J. A Novel Eighth-Order Hyperchaotic System and Its Application in Image Encryption. *Mathematics* **2023**, *11*, 4099. [CrossRef]
4. Li, M.; Fang, X.; Ernest, A. A Color Image Encryption Method Based on Dynamic Selection Chaotic System and Singular Value Decomposition. *Mathematics* **2023**, *11*, 3274. [CrossRef]
5. Li, S.-Y.; Gai, Y.; Shih, K.-C.; Chen, C.-S. An Efficient Image Encryption Algorithm Based on Innovative DES Structure and Hyperchaotic Keys. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *70*, 4103–4111. [CrossRef]
6. Sahin, M.E. Memristive chaotic system-based hybrid image encryption application with AES and RSA algorithms. *Phys. Scr.* **2023**, *98*, 075216. [CrossRef]
7. Ding, Y.; Wang, Z.; Qin, Z.; Zhou, E.; Zhu, G.; Qin, Z.; Choo, K.-K.R. Backdoor Attack on Deep Learning-Based Medical Image Encryption and Decryption Network. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 280–292. [CrossRef]
8. Ding, C.; Xue, R. Signal-sensing dynamic S-box image encryption with 2D Griewank–sin map. *Nonlinear Dyn.* **2023**, *111*, 22595–22620. [CrossRef]
9. Chen, W.-H.; Zhou, X.-F.; Li, M.-J.; Hu, M. Image encryption algorithm based on optical chaos and elliptic curve. *Eur. Phys. J. D* **2023**, *77*, 197. [CrossRef]
10. Wang, M.; Wang, X.; Wang, C.; Zhou, S.; Xia, Z.; Li, Q. Color image encryption based on 2D enhanced hyperchaotic logistic-sine map and two-way Josephus traversing. *Digit. Signal Process.* **2023**, *132*, 103818. [CrossRef]
11. Lone, P.N.; Mir, U.H.; Gaffar, A. Hyperchaotic image encryption using DNA coding and discrete cosine transform. *J. Frankl. Inst.* **2023**, *360*, 13318–13338. [CrossRef]
12. Wang, Y.; Chen, J.; Wang, J. Visually meaningful image encryption based on 2D compressive sensing and dynamic embedding. *J. Inf. Secur. Appl.* **2023**, *78*, 103613. [CrossRef]
13. Mohamed, N.A.E.-S.; El-Sayed, H.; Youssif, A. Mixed Multi-Chaos Quantum Image Encryption Scheme Based on Quantum Cellular Automata (QCA). *Fractal Fract.* **2023**, *7*, 734. [CrossRef]
14. Chen, H.; Liu, Z.; Zhu, L.; Tanougast, C.; Blondel, W. Asymmetric color cryptosystem using chaotic Ushiki map and equal modulus decomposition in fractional Fourier transform domains. *Opt. Lasers Eng.* **2019**, *112*, 7–15. [CrossRef]
15. Wang, M.; Wang, X.; Zhao, T.; Zhang, C.; Xia, Z.; Yao, N. Spatiotemporal chaos in improved cross coupled map lattice and its application in a bit-level image encryption scheme. *Inf. Sci.* **2020**, *544*, 1–24. [CrossRef]
16. Wang, X.-Y.; Li, Z.-M. A color image encryption algorithm based on Hopfield chaotic neural network. *Opt. Lasers Eng.* **2018**, *115*, 107–118. [CrossRef]
17. Gayathri, J.; Subashini, S. An efficient spatiotemporal chaotic image cipher with an improved scrambling algorithm driven by dynamic diffusion phase. *Inf. Sci.* **2019**, *489*, 227–254. [CrossRef]
18. Wang, X.; Gao, S. Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. *Inf. Sci.* **2020**, *507*, 16–36. [CrossRef]

19. Wang, X.; Feng, L.; Zhao, H. Fast image encryption algorithm based on parallel computing system. *Inf. Sci.* **2019**, *486*, 340–358. [CrossRef]
20. Jamal, S.S.; Hazzazi, M.M.; Khan, M.F.; Bassfar, Z.; Aljaedi, A.; Islam, Z.U. Region of interest-based medical image encryption technique based on chaotic S-boxes. *Expert Syst. Appl.* **2023**, *238*, 122030. [CrossRef]
21. Zhou, S.; Wang, X.; Wang, Z.; Zhang, C. A novel method based on the pseudo-orbits to calculate the largest Lyapunov exponent from chaotic equations. *Chaos Interdiscip. J. Nonlinear Sci.* **2019**, *29*, 033125. [CrossRef] [PubMed]
22. Teng, L.; Wang, X.; Xian, Y. Image encryption algorithm based on a 2D-CLSS hyperchaotic map using simultaneous permutation and diffusion. *Inf. Sci.* **2022**, *605*, 71–85. [CrossRef]
23. Wang, X.; Liu, P. A New Full Chaos Coupled Mapping Lattice and Its Application in Privacy Image Encryption. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *69*, 1291–1301. [CrossRef]
24. Wang, M.; Wang, X.; Wang, C.; Xia, Z.; Zhao, H.; Gao, S.; Zhou, S.; Yao, N. Spatiotemporal chaos in cross coupled map lattice with dynamic coupling coefficient and its application in bit-level color image encryption. *Chaos Solitons Fractals* **2020**, *139*, 110028. [CrossRef]
25. 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]
26. Zhong, H.; Li, G.; Xu, X.; Song, X. Image Encryption Algorithm Based on a Novel Wide-Range Discrete Hyperchaotic Map. *Mathematics* **2022**, *10*, 2583. [CrossRef]
27. Shao, Z.; Wang, X.; Tang, Y.; Shang, Y. Trinion discrete cosine transform with application to color image encryption. *Multimed. Tools Appl.* **2023**, *82*, 14633–14659. [CrossRef]
28. Feng, W.; Zhao, X.; Zhang, J.; Qin, Z.; Zhang, J.; He, Y. Image Encryption Algorithm Based on Plane-Level Image Filtering and Discrete Logarithmic Transform. *Mathematics* **2022**, *10*, 2751. [CrossRef]
29. Kaur, G.; Agarwal, R.; Patidar, V. Double Image Encryption Based on 2D Discrete Fractional Fourier Transform and Piecewise Nonlinear Chaotic Map. *Adv. Inform. Comput. Res.* **2019**, *955*, 519–530. [CrossRef]
30. Qiu, T.; Dai, W.-H.; Chen, S.-H.; Zhou, H.; Gong, L.-H. Double-image encryption algorithm based on discrete fractional angular transform and fractional Fourier transform. *Opt. Appl.* **2022**, *52*, 669–684. [CrossRef]
31. Shao, Z.; Tang, Y.; Liang, M.; Shang, Y.; Wang, F.; Wang, Y. Double image encryption based on symmetry of 2D-DFT and equal modulus decomposition. *Multimed. Tools Appl.* **2021**, *80*, 8973–8998. [CrossRef]
32. Tong, L.-J.; Zhou, N.-R.; Huang, Z.-J.; 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**, *2021*, 6650515. [CrossRef]
33. Tang, Y.; Shao, Z.; Zhao, X.; Shang, Y. Robust multiple color images encryption using discrete Fourier transforms and chaotic map. *Signal Process. Image Commun.* **2021**, *93*, 116168. [CrossRef]
34. Wang, M.-M.; Zhu, C.-N.; Li, L.; Xu, M.-T. Double-image compression and encryption scheme based on chaotic system and real-preserving multi-order discrete fractional Fourier transform. *J. Mod. Opt.* **2022**, *69*, 917–930. [CrossRef]
35. Xiao, M.; Tan, R.; Ye, H.; Gong, L.; Zhu, Z. Double-Color-Image Compression-Encryption Algorithm Based on Quaternion Multiple Parameter DFrAT and Feature Fusion with Preferable Restoration Quality. *Entropy* **2022**, *24*, 941. [CrossRef]
36. Wang, X.Y.; Yang, L.; Liu, R.; Kadir, A. A chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn.* **2010**, *62*, 615–621. [CrossRef]
37. Chai, X.; Chen, Y.; Broyde, L. A novel chaos-based image encryption algorithm using DNA sequence operations. *Opt. Lasers Eng.* **2017**, *88*, 197–213. [CrossRef]
38. Ye, G.; Huang, X. A secure image encryption algorithm based on chaotic maps and SHA-3. *Secur. Commun. Netw.* **2016**, *9*, 2015–2023. [CrossRef]
39. Zhang, W.; Wang, S.; Han, W.; Yu, H.; Zhu, Z. An Image Encryption Algorithm Based on Random Hamiltonian Path. *Entropy* **2020**, *22*, 73. [CrossRef]
40. Zheng, J.; Luo, Z.; Zeng, Q. An efficient image encryption algorithm based on multi chaotic system and random DAN coding. *Multimed. Tools Appl.* **2020**, *79*, 29901–29921. [CrossRef]
41. Erkan, U.; Toktas, A.; Toktas, F.; Alenezi, F. 2D eπ-map for image encryption. *Inf. Sci.* **2022**, *589*, 770–789. [CrossRef]
42. Li, L.; Luo, Y.; Qiu, S.; Ouyang, X.; Cao, L.; Tang, S. Image encryption using chaotic map and cellular automata. *Multimed. Tools Appl.* **2022**, *81*, 40755–40773. [CrossRef]
43. Agarwal, V.; Kumar, D. Secure chaotic image encryption method using random graph traversal and three step diffusion. *Multimed. Tools Appl.* **2023**, 1–26. [CrossRef]
44. 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]