



Article

Image Encryption Algorithm Based on Plane-Level Image Filtering and Discrete Logarithmic Transform

Wei Feng ¹, Xiangyu Zhao ^{2,*}, Jing Zhang ¹, Zhentao Qin ¹, Junkun Zhang ¹ and Yigang He ³

¹ School of Mathematics and Computer Science, Panzhuhua University, Panzhuhua 617000, China; fengwei@pzhuhua.edu.cn (W.F.); zjpzh@tom.com (J.Z.); qinzhenhao@126.com (Z.Q.); bearzjk@sina.com.cn (J.Z.)

² School of Electrical and Information Engineering, Panzhuhua University, Panzhuhua 617000, China

³ School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China; yghe1221@whu.edu.cn

* Correspondence: zhaoxiangyu@pzhuhua.edu.cn

Abstract: Image encryption is an effective way to protect image data. However, existing image encryption algorithms are still unable to strike a good balance between security and efficiency. To overcome the shortcomings of these algorithms, an image encryption algorithm based on plane-level image filtering and discrete logarithmic transformation (IEA-IF-DLT) is proposed. By utilizing the hash value more rationally, our proposed IEA-IF-DLT avoids the overhead caused by repeated generations of chaotic sequences and further improves the encryption efficiency through plane-level and three-dimensional (3D) encryption operations. Aiming at the problem that common modular addition and XOR operations are subject to differential attacks, IEA-IF-DLT additionally includes discrete logarithmic transformation to boost security. In IEA-IF-DLT, the plain image is first transformed into a 3D image, and then three rounds of plane-level permutation, plane-level pixel filtering, and 3D chaotic image superposition are performed. Next, after a discrete logarithmic transformation, a random pixel swapping is conducted to obtain the cipher image. To demonstrate the superiority of IEA-IF-DLT, we compared it with some state-of-the-art algorithms. The test and analysis results show that IEA-IF-DLT not only has better security performance, but also exhibits significant efficiency advantages.

Keywords: image encryption; cryptanalysis; image filtering; discrete logarithm; security analysis

MSC: 94-08; 94A60



Citation: 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. <https://doi.org/10.3390/math10152751>

Academic Editor: Jakub Nalepa

Received: 1 July 2022

Accepted: 1 August 2022

Published: 3 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to the rapid rise and popularization of information technology, a huge amount of digital information is generated every second in the world today and spread through various channels. Among these different formats of digital information, digital images are especially widely used because they can convey information concisely and vividly. However, for reasons such as privacy protection, commercial security, and military security, it is desirable to provide these disseminated images with effective protection against unauthorized access [1–4]. Because of this, how to provide more secure and efficient protection for digital images has been the focus of researchers in recent years [5–8]. Among several protection methods, image encryption is increasingly preferred as a straightforward and effective one. After being processed by image encryption, digital images will become unrecognizable cipher images, similar to noise images. Without the correct secret key, unauthorized users cannot obtain useful information from these noise-like images. On the other hand, these unrecognizable cipher images can be decrypted into useful normal images when the correct secret key is applied. It is worth noting that digital images have many characteristics different from text data, such as high information redundancy and

strong correlation between adjacent pixels. This makes traditional encryption algorithms, including Data Encryption Standard (DES), Advanced Encryption Standard (AES), and Internationale Data Encrypt Algorithm (IDEA), not well suited to the encryption of digital images [9–11]. Therefore, in order to securely and effectively safeguard digital images, researchers are increasingly using new technologies and methodologies to design image encryption algorithms, such as chaotic systems [7,12,13], DNA computing [14–16], quantum computing [11,17,18], and compressed sensing [19–21].

Chaotic systems have many properties that conform to the design principles of cryptosystems, such as initial value sensitivity, ergodicity, and unpredictability. Many image encryption algorithms based on chaotic systems are proposed every year [1,3,5]. Through the coupling of logistic and tent maps, Hua et al. [22] constructed a two-dimensional (2D) map with excellent chaotic performance and further proposed a novel color image encryption algorithm based on this 2D map. In [23], after introducing two new one-dimensional (1D) chaotic systems, Wang et al. presented an image encryption algorithm employing dynamic row permutation and Zigzag transformation. Based on a chaotic map, Bezerra et al. [24] suggested an image encryption algorithm using a permutation and diffusion structure. In [25], exploiting a circular bit-level scrambling method, Diaconu presented a chaotic image encryption algorithm. Based on a hybrid model of DNA computing, chaotic systems, and hash functions, Zefreh [26] designed an image encryption algorithm that mainly includes two steps of DNA-level permutation and DNA-level diffusion. Benefiting from the unique characteristics of chaotic systems and the excellent randomness of the chaotic sequences generated by them, existing image encryption algorithms based on chaotic systems have demonstrated good encryption effects. To the best of our knowledge, there are no cases where these algorithms have been successfully broken by ciphertext-only attacks. However, based on our cryptanalysis research [27–31] and the work done by other researchers [3,5,32,33], we found that existing image encryption algorithms still have the following shortcomings.

- (1) The secret key design of some algorithms is not reasonable, resulting in the need to change the secret key every time a different image is encrypted. Such a design is not practical when there are a large number of images to be encrypted.
- (2) Inappropriate use of the plain image hash value. This makes it necessary to repeatedly generate chaotic sequences when encrypting different images.
- (3) Some algorithms only make simple use of modular addition or XOR operations, making them vulnerable to differential attacks.
- (4) Bit-level or pixel-by-pixel encryption operations make some algorithms inefficient at encrypting images.

In this research, to address the shortcomings of existing algorithms, we propose an image encryption algorithm based on plane-level image filtering and discrete logarithmic transform (IEA-IF-DLT). The following is a summary of the novelties and contributions of our proposed algorithm.

- (1) A standardized and reasonable secret key design is adopted, and there is no need to change the secret key when encrypting different images.
- (2) The hash value is used to truncate chaotic sequences and to determine the generator of the finite multiplicative group Z_{257}^* . This allows the chaotic sequences to be generated in advance and reused once the secret key is determined.
- (3) A discrete logarithmic transformation based on Z_{257}^* is employed, thereby rendering common differential attack strategies ineffective.
- (4) The plane-level permutation, plane-level image filtering, and three-dimensional (3D) chaotic image superposition make the encryption efficiency extremely high while ensuring security.
- (5) The random pixel swapping performed at the end makes it impossible for attackers to isolate the diffusion operation through special plain images.

The rest of this paper is structured as follows. In Section 2, the chaotic systems, SHA-256 hash value, and discrete logarithm are introduced. In Section 3, our proposed IEA-IF-DLT is described in detail. In Section 4, simulation tests and related analyses are presented to verify the superiority of IEA-IF-DLT. In the last section, we conclude the work done in this paper.

2. Preliminaries

In our proposed IEA-IF-DLT, two chaotic systems are exploited to generate chaotic sequences, the SHA-256 hash function is employed to improve the plain image sensitivity of IEA-IF-DLT, and the discrete logarithm is utilized to enhance the non-linearity of the encryption process, thereby improving the security of the entire algorithm.

2.1. Chaotic Systems

In the related research of image encryption, compared with low-dimensional chaotic systems, high-dimensional chaotic systems have more complex dynamics, so it is favored by many researchers. In IEA-IF-DLT, we leverage the 4D hyper-chaotic Chen system introduced in [34]. Mathematically, this 4D hyper-chaotic system can be defined as follows.

$$\begin{cases} \dot{x} = \alpha(y - x), \\ \dot{y} = \gamma x - xz + \mu y - w, \\ \dot{z} = xy - \beta z, \\ \dot{w} = x + \lambda, \end{cases} \tag{1}$$

where x, y, z, w are four system state variables, and $\alpha, \beta, \mu, \gamma, \lambda$ are five system control parameters. When $(\alpha, \beta, \mu, \gamma) = (36, 3, 28, -16)$ and $\lambda \in [-0.7, 0.7]$, this system is hyper-chaotic. Figure 1 demonstrates the rich dynamic features of this 4D hyper-chaotic system, which makes it an excellent choice for image encryption.

In addition to high-dimensional continuous chaotic systems, the study of discrete chaotic maps has also attracted the attention of many researchers [22,35,36]. In [22], to overcome the shortcomings of existing chaotic maps, Hua et al. proposed a two-dimensional logistic tent modular map (2D-LTMM). This newly reported chaotic map has a wide and continuous chaotic range and more evenly distributed trajectories, making it ideal for image encryption. Specifically, the 2D-LTMM has the following mathematical definition.

$$\begin{cases} x_{i+1} = \begin{cases} (4r_1x_i(1 - x_i) + 2r_2y_i) \bmod 1 & \text{when } y_i < 0.5, \\ (4r_1x_i(1 - x_i) + 2r_2(1 - y_i)) \bmod 1 & \text{when } y_i \geq 0.5, \end{cases} \\ y_{i+1} = \begin{cases} (4r_1y_i(1 - y_i) + 2r_2x_i) \bmod 1 & \text{when } x_i < 0.5, \\ (4r_1y_i(1 - y_i) + 2r_2(1 - x_i)) \bmod 1 & \text{when } x_i \geq 0.5, \end{cases} \end{cases} \tag{2}$$

where x_i and y_i are the outputs of the i -th iteration as well as the inputs of the $(i + 1)$ -th iteration, while r_1 and r_2 are the control parameters of 2D-LTMM. When $r_1, r_2 \in [1, 100]$, the 2D-LTMM is in a hyperchaotic state.

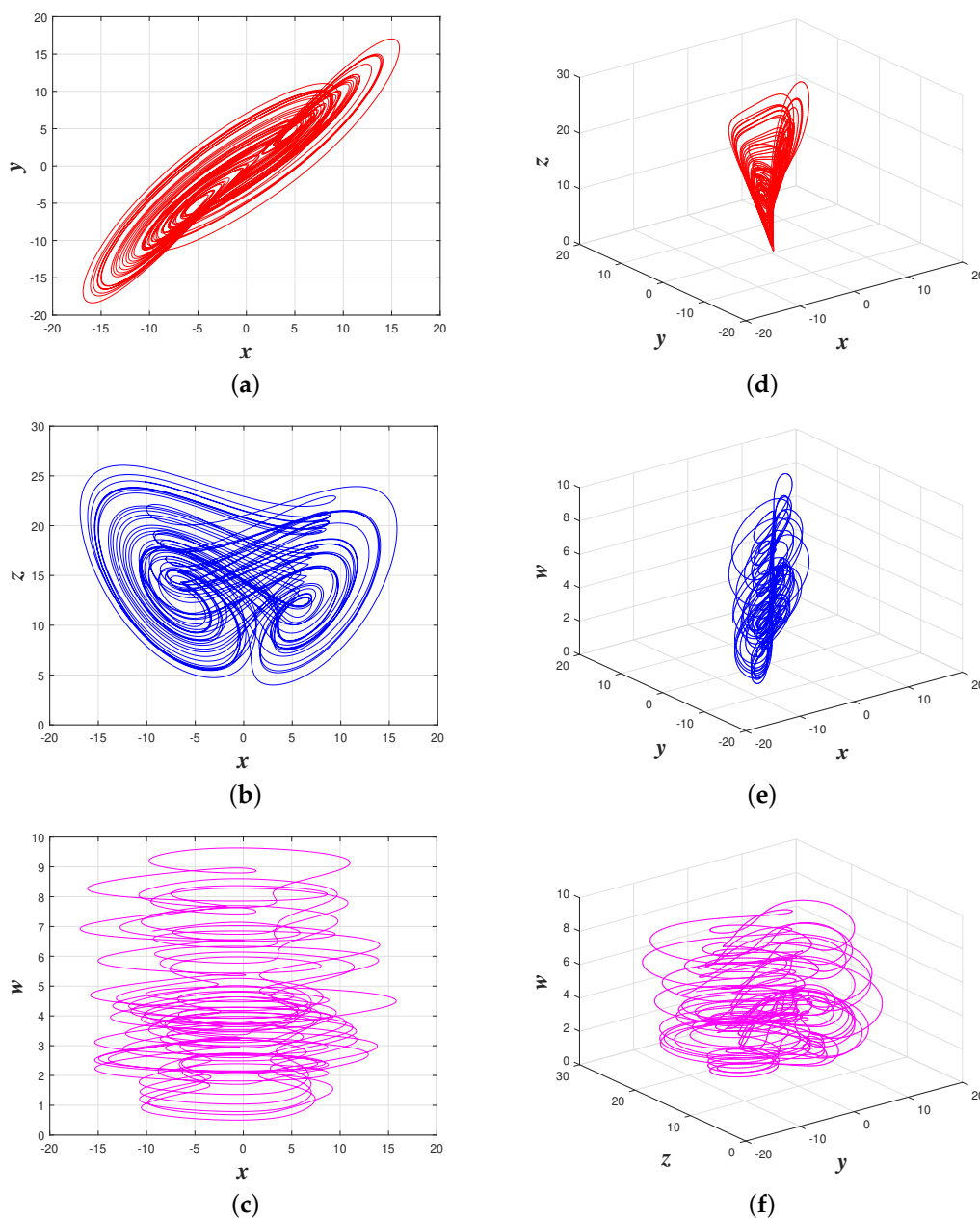


Figure 1. Six hyper-chaotic attractors of adopted hyper-chaotic Chen system: (a) x-y plane; (b) x-z plane; (c) x-w plane; (d) x-y-z plane; (e) x-y-w plane; (f) y-z-w plane.

2.2. SHA-256 Hash Value

The SHA-256 hash function was released by the National Institute of Standards and Technology (NIST) in 2001. Due to its outstanding performance, SHA-256 is now widely used in numerous fields, such as blockchain. For any input message with a length of less than $2^{64} - 1$ bits, SHA-256 can generate a hash value with a fixed length of 256 bits. Furthermore, SHA-256 is extremely sensitive to input—even a single bit change in the input message can drastically alter the hash value generated. Because of its remarkable input sensitivity, many image encryption algorithms incorporate SHA-256 to boost their plain image sensitivity [23,24,37,38]. However, it is worth mentioning that these algorithms employ the plain image hash value in an unreasonable way, either directly as a secret key or to produce the parameters of chaotic systems. Both of these pose a practical problem; that is, when there are a large number of images to be encrypted, users must constantly change secret keys or iterate the chaotic systems constantly to obtain the desired key streams. As a

result, in our suggested IEA-IF-DLT, we improve the method of utilizing the plain image hash value. Specifically, for the the plain image P of size $M \times N$, the hash value of P is first split into 32 bytes, which are h_1, h_2, \dots, h_{32} . Then, two parameters to be used in the encryption process are generated,

$$\begin{cases} H^{(1)} = \left(\sum_{i=1}^{32} h_i \right) \bmod M, \\ H^{(2)} = \left(\sum_{i=1}^{32} h_i \right) \bmod N. \end{cases} \tag{3}$$

2.3. Discrete Logarithm

For a prime integer p , under the modulo- p multiplication operation, we can construct the modulo- p multiplicative group $Z_p^* = \{1, 2, \dots, p - 1\}$. Furthermore, discrete logarithms can be defined over Z_p^* . If $a = g^b \bmod p$ holds, where g is a generator of Z_p^* , then b is referred to as the discrete logarithm of a , denoted by $b = \log_g a \bmod p$ [14,39]. Considering the characteristics of image encryption, in our proposed IEA-IF-DLT, we adopt the discrete logarithmic operation based on the finite multiplicative group Z_{257}^* . In contrast to the modular addition and XOR operations widely used in existing image encryption algorithms, the discrete logarithmic operation is a sort of non-linear operation. Therefore, applying the discrete logarithmic operation to image encryption can make the mathematical relationship between plain pixels and cipher pixels more complicated, thus making common differential attack strategies ineffective. Since the solution of discrete logarithms is a complex mathematical problem, directly performing discrete logarithmic operations in IEA-IF-DLT will undoubtedly reduce its encryption efficiency. To solve this problem, we calculate all the discrete logarithms of each $a \in Z_{257}^*$ under different generators in advance and then store them in the 2D matrix Θ of size 128×256 . In this way, the IEA-IF-DLT can retrieve the results of discrete logarithmic operations immediately by accessing the matrix Θ , rather than actually doing the complex discrete logarithmic calculations. In the matrix Θ , the row index represents the generator under which the discrete logarithm is calculated, and the column index represents the operand whose discrete logarithm needs to be calculated. The relationship between row indices and generators can be seen in Table 1.

Table 1. First 64 generators of Z_{257}^* .

Row Index (Generator g)							
1 (3)	2 (5)	3 (6)	4 (7)	5 (10)	6 (12)	7 (14)	8 (19)
9 (20)	10 (24)	11 (27)	12 (28)	13 (33)	14 (37)	15 (38)	16 (39)
17 (40)	18 (41)	19 (43)	20 (45)	21 (47)	22 (48)	23 (51)	24 (53)
25 (54)	26 (55)	27 (56)	28 (63)	29 (65)	30 (66)	31 (69)	32 (71)
33 (74)	34 (75)	35 (76)	36 (77)	37 (78)	38 (80)	39 (82)	40 (83)
41 (85)	42 (86)	43 (87)	44 (90)	45 (91)	46 (93)	47 (94)	48 (96)
49 (97)	50 (101)	51 (102)	52 (103)	53 (105)	54 (106)	55 (107)	56 (108)
57 (109)	58 (110)	59 (112)	60 (115)	61 (119)	62 (125)	63 (126)	64 (127)

3. Proposed Encryption Algorithm

As stated in Section 1, to address some shortcomings in some current image encryption algorithms, we suggest an image encryption algorithm called IEA-IF-DLT. The essential components of our suggested IEA-IF-DLT are following encryption steps: determination of three dimensions, generation of chaotic sequences, plane-level permutation, plane-level image filtering, 3D chaotic image superposition, discrete logarithmic transformation, and random pixel swapping, as illustrated in Figure 2. To enhance security and maximize the efficiency advantage of plane-level operations, among them, plane-level permutation, plane-level image filtering, and 3D chaotic image superposition are iterated for three rounds in different forms. Below, we go into depth about each encryption step.

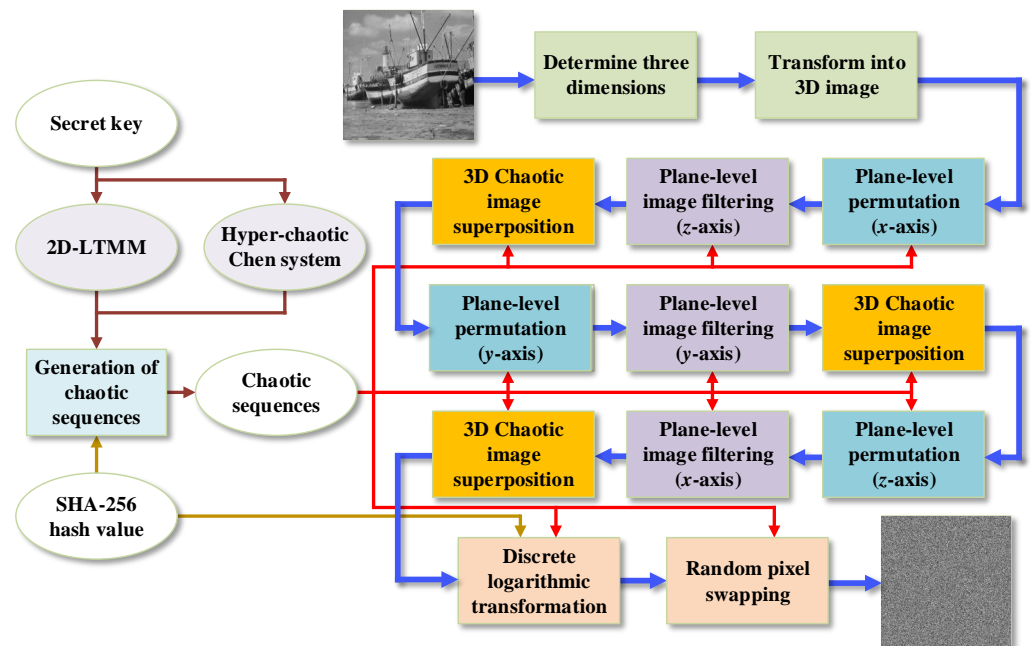


Figure 2. Flowchat of IEA-IF-DLT.

3.1. Determination of Three Dimensions

To begin, we need to change the representation of the plain image P from 2D to 3D. For the 2D image P of size $M \times N$, the following steps are devised to transform it into the intermediate cipher image $C^{(1)}$ in 3D form.

- **Step 1:** Determine the total number $t^{(P)}$ of the pixels in P . If $t^{(P)}$ is not a power of 2 or less than 8, fill P with the pixels whose values are zeros, until $t^{(P)}$ is a power of 2 and not less than 8. Otherwise, go to the next step.
- **Step 2:** According to $t^{(P)}$, calculate

$$\phi = \lfloor (\log_2 t^{(P)}) / 3 \rfloor, \tag{4}$$

where $\lfloor \bullet \rfloor$ represents the round down operation on an operand.

- **Step 3:** Determine the size $d^{(1)}$ of the first dimension, let $d^{(1)} = 2^\phi$.
- **Step 4:** For the size $d^{(2)}$ of the second dimension, let $d^{(2)} = d^{(1)}$.
- **Step 5:** According to $t^{(P)}$, $d^{(1)}$, and $d^{(2)}$, calculate the size

$$d^{(3)} = 2^{\log_2 t^{(P)} - d^{(1)} - d^{(2)}} \tag{5}$$

of the third dimension.

- **Step 6:** Reshape P into $C^{(1)}$ with the size of $d^{(1)} \times d^{(2)} \times d^{(3)}$.

Following the steps above, for a test image of size 256×256 , it will be transformed into the 3D form of size $32 \times 32 \times 64$. Similarly, two other common sizes, 512×512 and 1024×1024 , will be transformed into $64 \times 64 \times 64$ and $64 \times 64 \times 256$, respectively. Please keep in mind that, for the sake of clarity and convenience, this paper solely covers the case where the plain image is not padded. In other words, it is assumed that $d^{(1)} \times d^{(2)} \times d^{(3)} = M \times N$ is always true.

3.2. Generation of Chaotic Sequences

As described in Section 2.1, our proposed IEA-IF-DLT leverages chaotic sequences to encrypt the plain image, and these chaotic sequences are generated by the hyper-chaotic Chen system and 2D-LTMM. Firstly, according to Equations (18) and (19), the secret key K is converted into the initial state values and control parameters of the two chaotic system.

Then, together with the other fixed system control parameters, they are input into the chaotic systems to generate the chaotic sequence $S^{(1)}$ and $S^{(2)}$. More specifically, $S^{(1)}$ is generated by the 2D-LTMM, and its length is

$$L^{(1)} = H^{(1)} + \sigma^{(d)}, \tag{6}$$

where $H^{(1)}$ is defined in Section 2.2, $\sigma^{(d)} = d^{(1)} + d^{(2)} + d^{(3)}$, and $d^{(1)}, d^{(2)}, d^{(3)}$ represent the sizes of the three dimensions, which are also already given in Section 3.1. $S^{(2)}$ is generated by the hyper-chaotic Chen system, and its length is

$$L^{(2)} = \sigma^{(d)} + 4\pi^{(d)} + 3, \tag{7}$$

where $\pi^{(d)} = d^{(1)} \times d^{(2)} \times d^{(3)}$. Finally, $S^{(1)}$ and $S^{(2)}$ are transformed into ten chaotic sequences $\hat{S}^{(1)}, \hat{S}^{(2)}, \hat{S}^{(3)}, \hat{S}^{(4)}, \hat{S}^{(5)}, \hat{S}^{(6)}, \hat{S}^{(7)}, \hat{S}^{(8)}, \hat{S}^{(9)}, \hat{S}^{(10)}$ used in subsequent encryption steps as follows.

$$\begin{cases} \hat{S}^{(1)} = S^{(1)}(H^{(1)} + 1 : H^{(1)} + d^{(1)}), \\ \hat{S}^{(4)} = S^{(1)}(H^{(1)} + d^{(1)} + 1 : H^{(1)} + d^{(1)} + d^{(2)}), \\ \hat{S}^{(7)} = S^{(1)}(H^{(1)} + d^{(1)} + d^{(2)} + 1 : H^{(1)} + \sigma^{(d)}), \end{cases} \tag{8}$$

$$\begin{cases} \hat{S}^{(2)} = \left\lfloor \left\lfloor S^{(2)}(1 : d^{(3)} + 1) \right\rfloor \times 10^{15} \right\rfloor \bmod 256, \\ \hat{S}^{(5)} = \left\lfloor \left\lfloor S^{(2)}(d^{(3)} + \pi^{(d)} + 2 : d^{(2)} + d^{(3)} + \pi^{(d)} + 2) \right\rfloor \times 10^{15} \right\rfloor \bmod 256, \\ \hat{S}^{(8)} = \left\lfloor \left\lfloor S^{(2)}(d^{(2)} + d^{(3)} + 2\pi^{(d)} + 3 : \sigma^{(d)} + 2\pi^{(d)} + 3) \right\rfloor \times 10^{15} \right\rfloor \bmod 256, \end{cases} \tag{9}$$

$$\begin{cases} \hat{S}^{(3)} = \left\lfloor \left\lfloor S^{(2)}(d^{(3)} + 2 : d^{(3)} + \pi^{(d)} + 1) \right\rfloor \times 10^{15} \right\rfloor \bmod 256, \\ \hat{S}^{(6)} = \left\lfloor \left\lfloor S^{(2)}(d^{(2)} + d^{(3)} + \pi^{(d)} + 3 : d^{(2)} + d^{(3)} + 2\pi^{(d)} + 2) \right\rfloor \times 10^{15} \right\rfloor \bmod 256, \\ \hat{S}^{(9)} = \left\lfloor \left\lfloor S^{(2)}(\sigma^{(d)} + 2\pi^{(d)} + 4 : \sigma^{(d)} + 3\pi^{(d)} + 3) \right\rfloor \times 10^{15} \right\rfloor \bmod 256, \end{cases} \tag{10}$$

$$\hat{S}^{(10)} = \left(\left\lfloor \left\lfloor S^{(2)}(\sigma^{(d)} + 3\pi^{(d)} + 4 : \sigma^{(d)} + 4\pi^{(d)} + 3) \right\rfloor \times 10^{15} \right\rfloor \bmod \pi^{(d)} \right) + 1, \tag{11}$$

where $\hat{S}^{(1)}, \hat{S}^{(4)}, \hat{S}^{(7)}$ will be used for the plane-level permutation operations described in Section 3.3; $\hat{S}^{(2)}, \hat{S}^{(5)}, \hat{S}^{(8)}$ will be used for the plane-level image filtering operations described in Section 3.4; $\hat{S}^{(3)}, \hat{S}^{(6)}, \hat{S}^{(9)}$ will be used to generate the chaotic pixel matrices described in Section 3.5, and $\hat{S}^{(10)}$ will be used for random pixel swapping described in Section 3.7.

3.3. Plane-Level Permutation

When compared to pixel-by-pixel permutation methods, vector-level permutation methods can improve encryption efficiency while maintaining the effect of confusion [40,41]. As shown in Figure 2, in order to quickly scramble the pixels in the intermediate cipher image $C^{(1)}$, our proposed IEA-IF-DLT performs three rounds of plane-level permutation operations from the x -axis, y -axis, and z -axis directions, respectively. More specifically, in the plane-level permutation along the x -axis, IEA-IF-DLT first sorts the chaotic sequence $\hat{S}^{(1)}$ to obtain the index vector $I^{(x)}$. Then, according to $I^{(x)}$, the y - z planes of $C^{(1)}$ are scrambled along the x -axis. Figure 3 presents a simple example of the plane-level permutation along the x -axis.

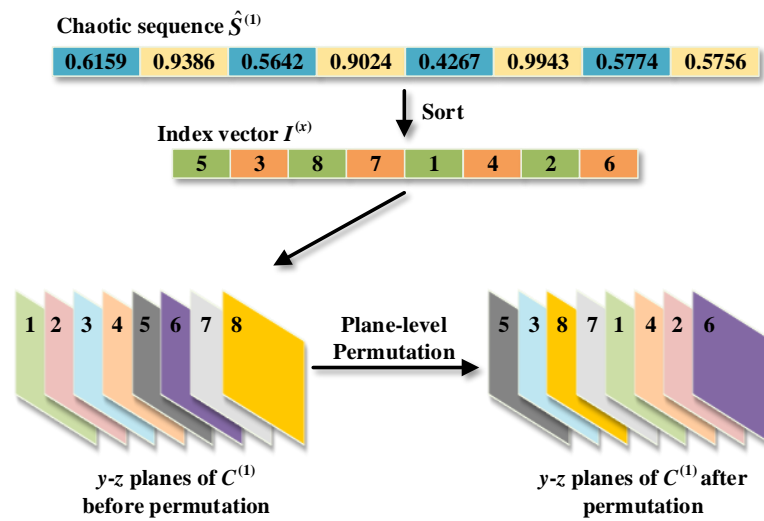


Figure 3. Plane-level permutation of IEA-IF-DLT along x-axis.

In this example, $\hat{S}^{(1)} = (0.6159, 0.9368, 0.5642, 0.9024, 0.4267, 0.9943, 0.5774, 0.5756)$ is first sorted in ascending order to obtain $I^{(x)} = (5, 3, 8, 7, 1, 4, 2, 6)$, then the eight y - z planes of $C^{(1)}$ are scrambled along the x -axis according to $I^{(x)}$. Obviously, scrambling the y - z planes can only change the relative positional relationship between pixels in the x -axis direction, but cannot change the relative positional relationship between pixels in each y - z plane. Consequently, after scrambling the y - z planes along the x -axis, our proposed IEA-IF-DLT then similarly scrambles the x - z planes and x - y planes in turn. The difference is that when scrambling the x - z planes along the y -axis, IEA-IF-DLT uses the sorting result of $\hat{S}^{(4)}$ for permutation, and when scrambling the x - y planes, the sorting result of $\hat{S}^{(7)}$ is utilized.

3.4. Plane-Level Image Filtering

Image filtering is a popular image processing technique that is commonly used in applications such as image smoothing, noise removal, and edge detection. Because traditional image filtering is irreversible, it cannot be applied directly to image encryption. However, with appropriate adjustments, some researchers have applied it to the diffusion operation of image encryption [42,43]. Different from the classic pixel diffusion method, image filtering can simultaneously diffuse multiple pixels to the current pixel in the form of convolution. Figure 4 shows the basic principle of pixel-level image filtering. In this simple example, it is the pixel p_9 that currently needs to be filtered. Pixel-level image filtering utilizes a filter mask of size 3×3 to diffuse adjacent pixels $p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8$ to p_9 , so as to obtain

$$c_9 = \left(\left(\sum_{i=1}^8 k_i \times p_i \right) + 1 \times p_9 \right) \bmod 256, \tag{12}$$

where $k_1, k_2, k_3, k_4, k_5, k_6, k_7$, and k_8 are the convolution coefficients of the filter mask. Likewise, in the reverse image filtering operation, one can restore c_9 to p_9 with the filter mask.

$$p_9 = \left(c_9 - \sum_{i=1}^8 k_i \times p_i \right) \bmod 256. \tag{13}$$

As can be seen, although the diffusion effect of pixel-level image filtering is relatively better, it is still a pixel-by-pixel diffusion method. Therefore, to further improve the efficiency of diffusion process, a novel plane-level image filtering method is devised in our proposed IEA-IF-DLT. Figure 5 presents an example of plane-level image filtering along the z -axis. In the demonstrated example, it is the x - y plane $P(:, :, 5)$ that currently needs to be

filtered. Plane-level image filtering exploits a filter mask of size 1×5 to diffuse adjacent planes $P(:, :, 1), P(:, :, 2), P(:, :, 3), P(:, :, 4)$ to $P(:, :, 5)$, so as to obtain

$$C(:, :, 5) = \left(\left(\sum_{i=1}^4 k_i \times P(:, :, i) \right) + 1 \times P(:, :, 5) \right) \bmod 256, \tag{14}$$

where k_1, k_2, k_3 , and k_4 are the convolution coefficients of the filter mask. Similarly, utilizing the same filter mask, one can restore $C(:, :, 5)$ to $P(:, :, 5)$ in the reverse image filtering operation.

$$P(:, :, 5) = \left(C(:, :, 5) - \sum_{i=1}^4 k_i \times P(:, :, i) \right) \bmod 256. \tag{15}$$

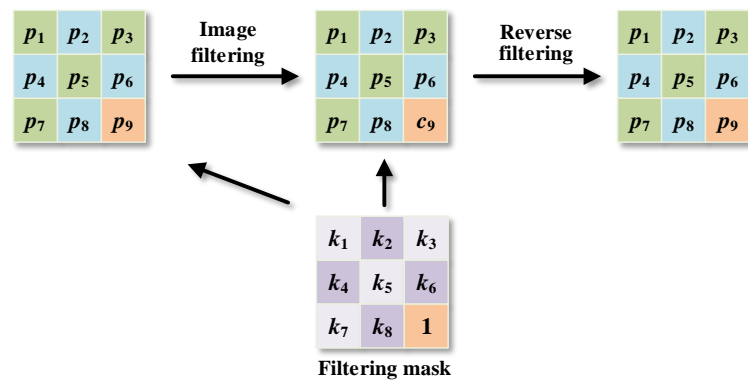


Figure 4. Basic principle of pixel-level image filtering.

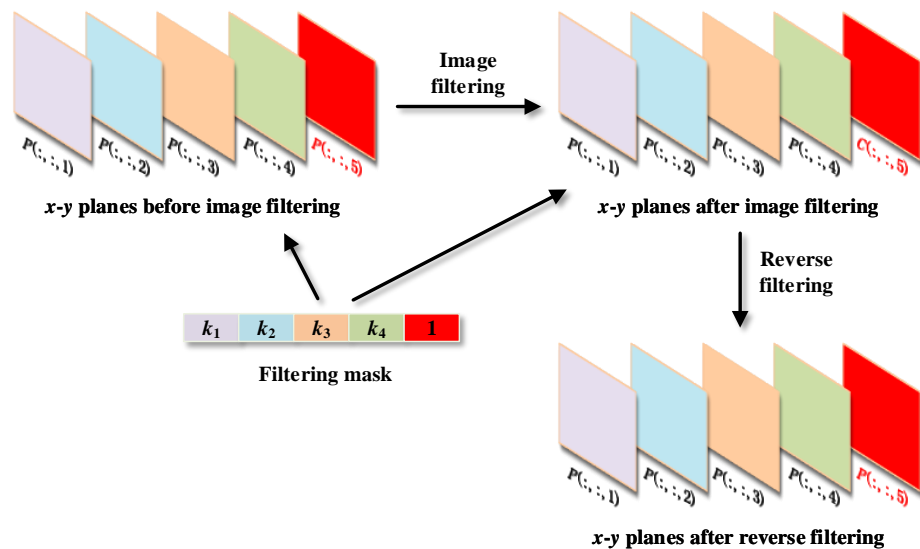


Figure 5. An example of plane-level image filtering.

Actually, in order to further enhance the sufficiency of pixel diffusion and thus ensure the plain image sensitivity of IEA-IF-DLT, we arranged three rounds of plane-level image filtering operations in IEA-IF-DLT, as shown in Figure 2. Among them, the first round of plane-level image filtering is to filter the x - y planes along the z -axis, and the used convolution coefficients come from the chaotic sequence $\hat{S}^{(2)}$. Next, the second round of plane-level image filtering is a filtering operation on the x - z planes along the y -axis, adopting the convolution coefficients from the chaotic sequence $\hat{S}^{(5)}$. Finally, the third round of plane-level image filtering processes the y - z planes along the x -axis, and adopts the convolution coefficients from the chaotic sequence $\hat{S}^{(8)}$.

3.5. 3D Chaotic Image Superposition

The randomness of chaotic sequences is exceedingly high. By superimposing chaotic pixels in the form of modular addition or XOR operations, image encryption algorithms can improve their security by increasing the randomness of cipher images and masking the statistical properties of plain images. As a result, many recent image encryption algorithms employ modular addition or XOR operations to superimpose chaotic pixels during the diffusion phase. However, there are two disadvantages to this methodology. First, pixel-by-pixel superimposing reduces the efficiency of the encryption process. Second, the single superimposing method is easily exploited by attackers, which leads to many image encryption algorithms being broken by them through chosen-plaintext attacks. To address these shortcomings, as illustrated in Figure 6, our proposed IEA-IF-DLT leverages three rounds of 3D chaotic image superposition.

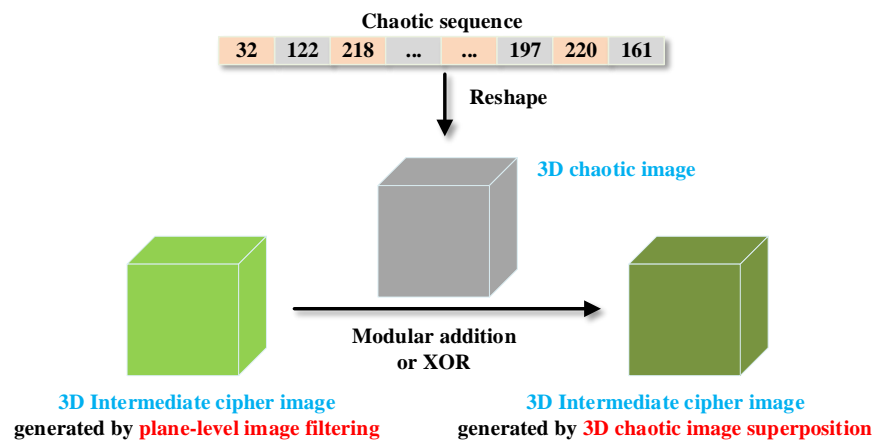


Figure 6. Chaotic image superposition of IEA-IF-DLT.

In the first round of 3D chaotic image superposition, the chaotic sequence $\hat{S}^{(3)}$ is reshaped into a 3D chaotic image, which is then superimposed on the intermediate cipher image created by plane-level image filtering in the manner of modular addition operation. In the second round of 3D chaotic image superposition, the chaotic sequence $\hat{S}^{(6)}$ is utilized to obtain a 3D chaotic image, and the chaotic image is superimposed on the intermediate cipher image in the form of XOR operation. The last round of 3D chaotic image superposition uses the chaotic sequence $\hat{S}^{(9)}$ to generate a 3D chaotic image, and then superimposes it in the form of modular addition operation. Because the three rounds of chaotic image superposition employ two distinct operations, the mathematical relationship between plain and cipher pixels can be more convoluted, thereby effectively resisting common plaintext attack methodologies.

3.6. Discrete Logarithmic Transformation

The discrete logarithm operation is a complex nonlinear operation that differs from the common modular addition and XOR operations used in some image encryption algorithms. To further improve the security of our proposed IEA-IF-DLT, we perform a discrete logarithmic transformation on the intermediate cipher image $\tilde{C}^{(1)}$ obtained after three rounds of plane-level permutation, plane-level image filtering, and 3D chaotic image superposition. The infinite multiplicative group Z_{257}^* , as mentioned in Section 2.3, contains up to 128 generators that can be exploited, and the discrete logarithms of the same value under different generators are significantly different. Table 2 provides the discrete logarithms of 9 under various generators. It is clear that when the generator differs, the discrete logarithm of 9 differs. Consequently, we must first identify the generator of the

discrete logarithm operation before applying the discrete logarithmic transformation. In IEA-IF-DLT, the generator g can be determined in the manner described below.

$$g = \left(\left(H^{(1)} \times H^{(2)} + \hat{S}^{(2)}(1) + \hat{S}^{(3)}(1) + \hat{S}^{(5)}(1) + \hat{S}^{(6)}(1) \right) \bmod 128 \right) + 1, \quad (16)$$

where $H^{(1)}$ and $H^{(2)}$ are parameters linked to the hash value of the plain image, as established in Section 2.2, while $\hat{S}^{(2)}, \hat{S}^{(3)}, \hat{S}^{(5)}, \hat{S}^{(6)}$ are the chaotic sequences given in Section 3.1. Since g is associated with both the hash value and the chaotic sequences, this design can improve not only the plain image sensitivity of IEA-IF-DLT but also its key sensitivity. Next, we can perform the discrete logarithmic transformation on the intermediate cipher image $\tilde{C}^{(1)}$ of size $d^{(1)} \times d^{(2)} \times d^{(3)}$ as follows. It is worth noting that Θ is the 2D matrix of size 128×256 that contains all the discrete logarithms of each $a \in \mathbb{Z}_{257}^*$ under different generators. The definition and usage of Θ is given in Section 2.3.

- **Step 1:** Initialize the 3D intermediate cipher image $\tilde{C}^{(2)}$ of size $d^{(1)} \times d^{(2)} \times d^{(3)}$, which is used to save the transformation result.
- **Step 2:** Set the index $i^{(x)}$ of the first dimension to 1.
- **Step 3:** Set the index $i^{(y)}$ of the second dimension to 1.
- **Step 4:** Let $\tilde{C}^{(2)}(i^{(x)}, i^{(y)}, :) = \Theta(g, \tilde{C}^{(1)}(i^{(x)}, i^{(y)}, :) + 1) - 1$.
- **Step 5:** For $i^{(y)} = 2$ to $d^{(2)}$, repeat **Step 4**.
- **Step 6:** For $i^{(x)} = 2$ to $d^{(1)}$, repeat **Step 3** to **Step 5**.

Table 2. Discrete logarithms of 9 $\in \mathbb{Z}_{257}^*$ under different generators.

Generator g (Discrete Logarithm)							
3 (2)	5 (14)	6 (162)	7 (250)	10(174)	12(66)	14 (154)	19(170)
20 (78)	24 (226)	27 (86)	28 (58)	33 (26)	37 (166)	38 (74)	39 (134)
40 (238)	41 (54)	43 (94)	45 (18)	47 (42)	48 (130)	51 (146)	53 (210)
54 (246)	55 (102)	56 (218)	63 (206)	65 (194)	66 (186)	69 (106)	71 (22)
74 (70)	75 (30)	76 (234)	77 (82)	78 (38)	80 (142)	82 (214)	83 (222)
85 (158)	86 (254)	87 (62)	90 (178)	91 (126)	93 (118)	94 (202)	96 (34)
97 (46)	101 (198)	102 (50)	103 (242)	105 (138)	106 (114)	107 (190)	108 (150)
109 (230)	110 (6)	112 (122)	115 (182)	119 (10)	125 (90)	126 (110)	127 (98)

3.7. Random Pixel Swapping

According to prior studies on cryptanalysis, the majority of chosen-plaintext attack algorithms initiate the attack by using the special plain images of single pixel values [31,44,45]. Because encryption steps that alter pixel positions, such as permutation operations, are ineffective for such special plain images, attackers can effectively isolate the diffusion operation, thereby establishing the groundwork for further differential analysis [30,46]. Therefore, to avoid this, an encryption step called random pixel swapping is added to the end of our proposed IEA-IF-DLT. In random pixel swapping, the pixels of the intermediate cipher image generated by the discrete logarithmic transform are randomly repositioned once more. In this way, the attackers' strategy of conducting differential attacks by exploiting the special plain images of single pixel values is rendered ineffective.

One straightforward instance of random pixel swapping is shown in Figure 7. In this example, the size of the 3D intermediate cipher image that requires random pixel swapping is $2 \times 2 \times 4$. Firstly, the 3D image is converted into a 2D intermediate cipher image of size 4×4 . Next, the 1D chaotic sequence of 1×16 is also converted into a 2D chaotic matrix of size 4×4 , which is exactly the same size as the intermediate cipher image to be processed. In a one-to-one correspondence, each chaotic matrix element controls the swapping operation of the intermediate cipher pixel with the same coordinate. That is, the first chaotic matrix element is in charge of the first intermediate cipher pixel, the second chaotic matrix element is in charge of the second intermediate cipher pixel, and so on. For instance, the intermediate cipher pixel with a pixel value of 196 at (1,1) is controlled by the

chaotic matrix element at (1,1) with a value of 14. Finally, each chaotic matrix element is turned into a 2D coordinate, and the corresponding intermediate cipher pixel is swapped based on this coordinate. Since the elements of the chaotic matrix are random, the generated 2D coordinates are also random. In this example, the first chaotic matrix element is turned into the coordinate (4,2), hence the first intermediate cipher pixel is swapped with the pixel at (4,2) with a value of 111. Likewise, the second chaotic matrix element is turned into the coordinate (3,4), and the second intermediate cipher pixel is swapped with the pixel with a value of 14 according to this coordinate.

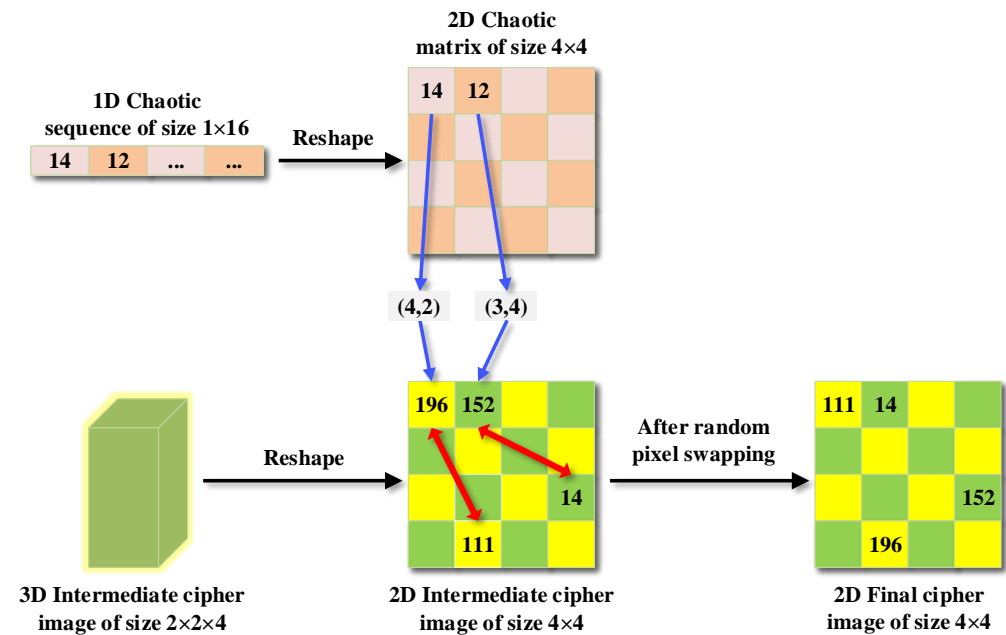


Figure 7. Random Pixel Swapping of IEA-IF-DLT.

More specifically, in our proposed IEA-IF-DLT, the random pixel swapping is performed as follows.

- **Step 1:** The 3D intermediate cipher image $\tilde{C}^{(2)}$ of size $d^{(1)} \times d^{(2)} \times d^{(3)}$ is reshaped into the 2D final cipher image C of size $M \times N$.
- **Step 2:** The 1D chaotic sequence $\hat{S}^{(10)}$ of length $1 \times (d^{(1)} \times d^{(2)} \times d^{(3)})$ is reshaped into the 2D chaotic matrix Φ of size $M \times N$.
- **Step 3:** Convert Φ into the random coordinate matrices $Y^{(r)}$ and $Y^{(c)}$ as follows.

$$\begin{cases} Y^{(c)} = ((\hat{S}^{(10)} - 1) \bmod N) + 1, \\ Y^{(r)} = \lfloor (\hat{S}^{(10)} - Y^{(c)}) / N \rfloor + 1, \end{cases} \quad (17)$$

where $Y^{(r)}$ holds the row numbers of the coordinates, and $Y^{(c)}$ holds the column numbers of the coordinates.

- **Step 4:** Set the row index $i^{(r)}$ to 1.
- **Step 5:** Set the column index $i^{(c)}$ to 1.
- **Step 6:** Swap $C(i^{(r)}, i^{(c)})$ with $C(Y^{(r)}(i^{(r)}, i^{(c)}), Y^{(c)}(i^{(r)}, i^{(c)}))$.
- **Step 7:** For $i^{(c)} = 2$ to N , repeat **Step 6**.
- **Step 8:** For $i^{(r)} = 2$ to M , repeat **Step 5** to **Step 7**.

Please note that, since our proposed IEA-IF-DLT is an image encryption algorithm with a symmetric structure, its decryption process is the inverse of the encryption process. For the sake of brevity, the description of decryption procedure is not repeated in this study.

3.8. Discussion

Because of the good randomness of chaotic sequences, most image encryption algorithms can ensure that the cipher images have good randomness. However, the main reason why many image encryption algorithms are cracked is that they cannot effectively resist differential attacks, especially chosen-plaintext attacks. Therefore, we have designed the following measures to ensure the security of our proposed IEA-IF-DLT.

- (1) The hash value of the plain image is used to truncate the chaotic sequences and determine the generator of the finite multiplicative group, which makes the equivalent key stream depend not only on the secret key but also on the plain image, thus helping to resist plaintext attacks.
- (2) Three rounds of plane-level permutation, plane-level image filtering, and 3D chaotic image superposition can realize good confusion and diffusion properties.
- (3) IEA-IF-DLT alternately uses modular addition and XOR operations to superimpose chaotic images, and the mathematical relationship between plain and cipher pixels becomes more complex, which helps to resist common plaintext attacks.
- (4) The discrete logarithmic operation is a complex nonlinear operation, which is different from the modular addition and XOR operations commonly used. Therefore, the discrete logarithmic transformation of the intermediate cipher image can enhance the ability of IEA-IF-DLT to resist plaintext attacks.
- (5) For the attackers' strategy of exploiting the special plain images of single pixel values to conduct differential attacks, the random pixel swapping added at the end of IEA-IF-DLT can effectively protect the previous encryption steps from being simplified or isolated.

As can be seen from the above measures, we do not rely solely on a single encryption step to ensure the security of IEA-IF-DLT. Our design idea is to ensure the security of IEA-IF-DLT through the targeted design of all encryption steps. To improve encryption efficiency, we adopt an S-box-like strategy to implement complex discrete logarithmic operations while employing a relatively small multiplicative group. However, it is worth noting that the purpose of introducing the discrete logarithmic transformation is not just to achieve the substitute effect it exhibits, but to ensure that there is an extremely complex mathematical relationship between plain and cipher pixels, together with other encryption steps, so as to effectively resist differential attacks. Next, in order to verify the advantages of IEA-IF-DLT in terms of security and efficiency, we conducted a large number of simulation tests and related analyses, and compared the test results with other advanced algorithms. As shown in Section 4, IEA-IF-DLT shows superiority in many security indicators, including information entropy.

4. Simulation Tests and Analyses

In order to comprehensively evaluate the performance and security of IEA-IF-DLT, we have completed a large number of simulation tests and related analyses, which include the visual effect test, key space analysis, key sensitivity analysis, differential attack analysis, histogram analysis, correlation analysis, information entropy analysis, robustness analysis, and efficiency analysis. These simulation tests and analyses were performed with the following hardware and software configurations: Intel(R) Xeon(R) CPU E3-1231 v3, 8 GB RAM, Window 7 operating system, and MATLAB R2017a (9.2.0538062). Besides, all test images are from The USC-SIPI Image Database (<http://sipi.usc.edu/database/>, accessed on 20 June 2022).

4.1. Visual Effect Test

A qualified image encryption algorithm must be able to transform plain images of different styles into unrecognizable random-like images. Without the correct secret key, no one can get any useful information from cipher images. However, once the correct secret key is used, one can fully recover the plain images. Figure 8 shows the relevant test results. After the processing of IEA-IF-DLT, the perceptible visual features of the plain

images are completely eliminated. Then, with the help of the correct secret key, random-like cipher images are turned into the plain images again without any loss of visual information. This means that, in a visual sense, the encryption and decryption effect of IEA-IF-DLT is qualified.

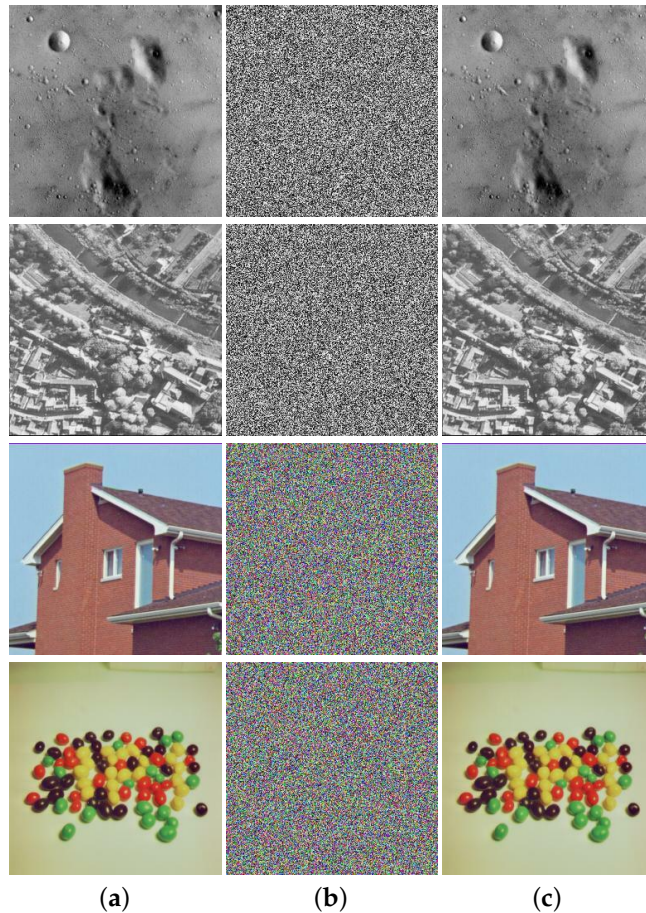


Figure 8. Results of visual effect test for IEA-IF-DLT: (a) plain images; (b) cipher images; (c) decrypted images.

4.2. Key Space Analysis

Some researchers have pointed out that, to effectively deal with brute force attacks, the key space of an image encryption algorithm should be at least greater than 2^{128} [5,32]. Considering the significant improvement in the computing power of computing hardware, we believe that this indicator should be increased to 2^{256} [14,15]. Furthermore, the representation of the secret key should also use a more canonical binary sequence form, otherwise it easily leads to the key sensitivity problem pointed out by Li et al [3,44]. In our proposed IEA-IF-DLT, the secret key is represented in the form of a binary sequence with the length of 260 bits, that is, $K = b_1 b_1 \dots b_{260}$. Specifically, the 260 bits of K will be converted into the initial state values and control parameters of the adopted chaotic systems in the following way.

$$\begin{cases} x_0^{(1)} = a_1 a_2 \dots a_{52} \times 2^{-52}, \\ y_0^{(1)} = a_{53} a_{54} \dots a_{104} \times 2^{-52}, \\ r_1^{(1)} = 1 + (a_{105} a_{106} \dots a_{156} + a_{157} a_{158} \dots a_{208}) \times 2^{-52}, \\ r_2^{(1)} = 1 + (a_{157} a_{158} \dots a_{208} + a_{209} a_{210} \dots a_{260}) \times 2^{-52}, \end{cases} \quad (18)$$

$$\left\{ \begin{array}{l} x_0^{(2)} = a_1 a_2 \dots a_{52} \times 2^{-52}, \\ y_0^{(2)} = a_{53} a_{54} \dots a_{104} \times 2^{-52}, \\ z_0^{(2)} = a_{105} a_{106} \dots a_{156} \times 2^{-52}, \\ w_0^{(2)} = a_{157} a_{158} \dots a_{208} \times 2^{-52}, \\ \lambda^{(2)} = -0.3 + a_{209} a_{210} \dots a_{260} \times 2^{-52}, \end{array} \right. \quad (19)$$

where $(x_0^{(1)}, y_0^{(1)})$, and $(r_1^{(1)}, r_1^{(2)})$ are the initial state values and control parameters of 2D-LTMM; $(x_0^{(2)}, y_0^{(2)}, z_0^{(2)}, w_0^{(2)})$ and $\lambda^{(2)}$ are the initial state values and control parameter of the hyper-chaotic Chen system. According to Equations (18) and (19), one can derive the key space of IEA-IF-DLT, $S = 2^{52} \times 2^{52} \times 2^{52} \times 2^{52} \times 2^{52} \times 2^{52} = 2^{260}$. Thus, IEA-IF-DLT has a large enough key space, and can effectively resist brute force attacks.

4.3. Key Sensitivity Analysis

For the design of a strong encryption algorithm, Claude Shannon suggests the concepts of confusion and diffusion. When it comes to image encryption, the confusion mentioned is to make the statistical relationship between the cipher image and the secret key as complex as possible, so that even if some statistical properties of the cipher image are obtained, attackers still cannot infer the secret key. In other words, a secure image encryption algorithm must have extremely high key sensitivity—even if the secret key changes only minimally, the resulting cipher image must change drastically. To evaluate the key sensitivity of our proposed IEA-IF-DLT, we randomly generated a secret key

$$K^{(1)} = D1801F11BAB61949DF8B2B33FA03B582 \\ EBA65B435F7433E2B08AA33F4A59AC5E6.$$

Then, we inverted the least significant bit of $K^{(1)}$, resulting in the secret key

$$K^{(2)} = D1801F11BAB61949DF8B2B33FA03B582 \\ EBA65B435F7433E2B08AA33F4A59AC5E7$$

that differs by only one bit. With these two secret keys with only one minimal difference, we obtained the key sensitivity test results for the encryption process, as shown in Figure 9. In the first row of the figure, the first image is the plain image 5.2.09, the second image is the cipher image obtained by encrypting 5.2.09 with $K^{(1)}$, the third image is the cipher image obtained by encrypting 5.2.09 with $K^{(2)}$, and the last image is the difference image of the first two cipher images; the pixel value distribution histograms of the images in the first row are shown in the second row. As one can see, using our proposed IEA-IF-DLT to encrypt the same plain image, even a small change in the secret key of only one bit will completely change the cipher image. Therefore, in the encryption process, IEA-IF-DLT has extremely high key sensitivity.

Similarly, we also tested the key sensitivity of IEA-IF-DLT during encryption, as shown in Figure 10. It can be seen that for the same cipher image, the plain image can be fully recovered if decrypted with the correct secret key. In contrast, the decrypted image looks like a noisy image even if the secret key changes only slightly. Therefore, in the decryption process, IEA-IF-DLT also has extremely high key sensitivity.

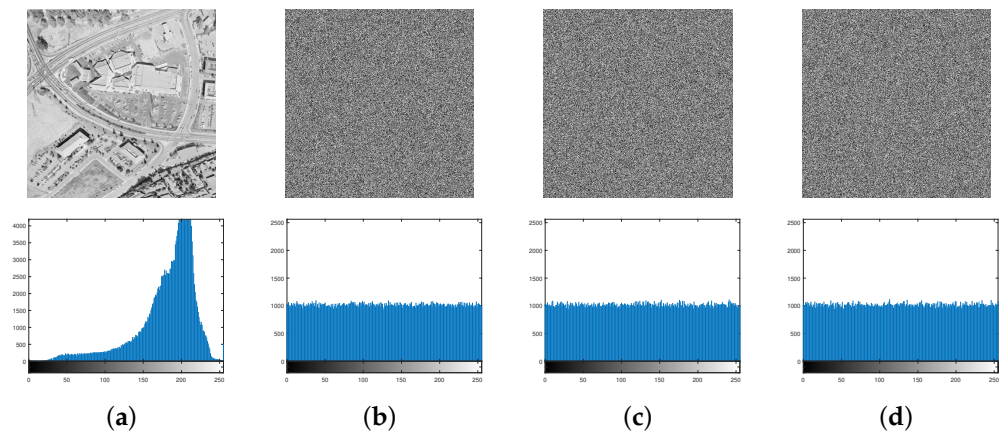


Figure 9. Key sensitivity test results for the encryption process: (a) 5.2.09; (b) cipher image $\tilde{C}^{(1)}$ obtained with $K^{(1)}$; (c) cipher image $\tilde{C}^{(2)}$ obtained with $K^{(2)}$; (d) Difference image obtained by $(\tilde{C}^{(2)} - \tilde{C}^{(1)}) \bmod 256$.

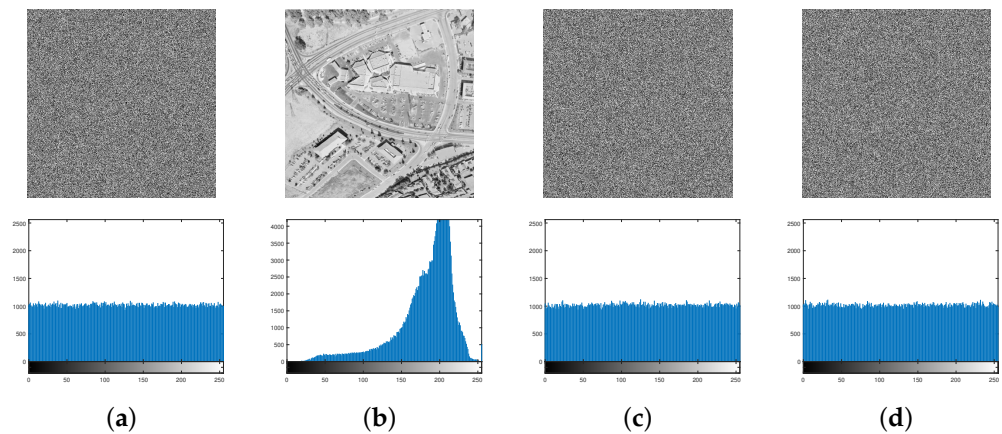


Figure 10. Key sensitivity test results for the decryption process: (a) cipher image $\tilde{C}^{(1)}$ obtained with $K^{(1)}$; (b) decrypted image $\tilde{D}^{(1)}$ obtained with $K^{(1)}$; (c) decrypted image $\tilde{D}^{(2)}$ obtained with $K^{(2)}$; (d) Difference image obtained by $(\tilde{D}^{(2)} - \tilde{D}^{(1)}) \bmod 256$.

4.4. Differential Attack Analysis

In the past few years, many image encryption algorithms have been cracked by attackers through differential attacks [27–31,44–46]. In order to reconstruct the plain image, differential attacks typically work by calculating and evaluating the changes in the cipher image brought on by little alterations to the plain image. Generally, one can exploit NPCR (number of pixels change rate) and UACI (unified average changing intensity) to quantitatively detect the ability of an encryption algorithm to resist differential attacks. These two metrics can be described mathematically as follows.

$$NPCR(I_1, I_2) = \sum_{x=1}^M \sum_{y=1}^N \frac{D(x,y)}{M \times N} \times 100\%, \tag{20}$$

$$UACI(I_1, I_2) = \sum_{x=1}^M \sum_{y=1}^N \frac{|I_1(x,y) - I_2(x,y)|}{255 \times M \times N} \times 100\%, \tag{21}$$

where I_1 and I_2 are two images whose difference needs to be evaluated, both have the size of $M \times N$; D represents the difference between I_1 and I_1 . When $I_1(x,y) \neq I_2(x,y)$, $D(x,y) = 1$, otherwise $D(x,y) = 0$. To evaluate the ability of our proposed IEA-IF-DLT to resist differential attacks, we tested it with 20 commonly used test images of different sizes, as shown in Tables 3 and 4. For each test image, we randomly changed one bit of it, and

then calculated the NPCR and UACI values between the cipher images before and after the change. As can be seen from Tables 3 and 4, the NPCR and UACI average of IEA-IF-DLT is closer to the optimum values of 99.6094% and 33.4635%, and the stability is better when compared to several recent image encryption algorithms.

Table 3. NPCR test results of IEA-IF-DLT and other image encryption algorithms.

Image Size	Filename	IEA-IF-DLT	Ref. [40]	Ref. [47]	Ref. [48]	Ref. [49]	Ref. [50]
256 × 256	5.1.09	99.6025	99.6136	99.6658	99.6292	99.6084	99.6140
	5.1.10	99.6094	99.6258	99.6475	99.6292	99.6155	99.5880
	5.1.11	99.6189	99.5787	99.6674	99.7055	99.6094	99.6033
	5.1.12	99.6178	99.6265	99.5941	99.7055	99.5758	99.5651
	5.1.13	99.5956	99.6246	99.6445	99.6765	99.6170	99.5789
512 × 512	5.1.14	99.6075	99.6134	99.5975	99.6765	99.6353	99.6765
	5.2.08	99.6136	99.6251	99.6281	99.6250	99.6151	99.6037
	5.2.09	99.5850	99.5703	99.6197	99.6292	99.6094	99.6029
	5.2.10	99.6181	99.6031	99.6288	99.6212	99.6166	99.6124
	7.1.01	99.6006	99.6124	99.6273	99.6208	99.5872	99.6082
	7.1.02	99.6170	99.6116	99.5892	99.6025	99.6109	99.6174
	boat.512	99.6178	99.6052	99.6006	99.6181	99.5998	99.6101
	elaine.512	99.6128	99.6131	99.6128	99.6076	99.6227	99.6087
	gray21.512	99.6052	99.6173	99.6082	99.6029	99.5949	99.6159
	numbers.512	99.6231	99.5912	99.6059	99.6081	99.6006	99.9075
1024 × 1024	ruler.512	99.6069	99.6168	99.6265	99.6033	99.6091	99.6212
	5.3.01	99.6082	99.6124	99.6098	99.6061	99.6035	99.6072
	5.3.02	99.6136	99.6231	99.6119	99.6190	99.6117	99.6116
	7.2.01	99.6128	99.6278	99.6156	99.6077	99.6013	99.6204
	testpat.1k	99.6037	99.6153	99.6124	99.6099	99.6048	99.6091
	Average	99.6095 ¹	99.6115	99.6207	99.6302	99.6075	99.6241
	Std. Dev.	0.00913 ¹	0.01565	0.02185	0.03311	0.01274	0.06993

¹ The bolded values emphasize that IEA-IF-DLT has the best performance in terms of average and stability.

Table 4. UACI test results of IEA-IF-DLT and other image encryption algorithms.

Image Size	Filename	IEA-IF-DLT	Ref. [40]	Ref. [47]	Ref. [48]	Ref. [49]	Ref. [50]
256 × 256	5.1.09	33.4823	33.4698	33.5980	33.3651	33.5253	33.4032
	5.1.10	33.4801	33.4425	33.5366	33.5240	33.5115	33.3557
	5.1.11	33.5077	33.3855	33.4398	33.5106	33.5174	33.4696
	5.1.12	33.4835	33.3982	33.4228	33.4172	33.4202	33.4634
	5.1.13	33.5054	33.5099	33.4205	33.5065	33.5019	33.3046
512 × 512	5.1.14	33.4667	33.3925	33.4696	33.4875	33.4939	33.4796
	5.2.08	33.4357	33.4410	33.4720	33.4973	33.4766	33.4493
	5.2.09	33.4687	33.4675	33.4921	33.4778	33.4528	33.5077
	5.2.10	33.4323	33.4502	33.4914	33.4327	33.3925	33.4457
	7.1.01	33.4514	33.5002	33.5212	33.4154	33.5017	33.4890
	7.1.02	33.4628	33.5121	33.4846	33.4698	33.4415	33.4190
	boat.512	33.4590	33.5100	33.5097	33.4472	33.4519	33.5414
	elaine.512	33.4593	33.4650	33.5477	33.4337	33.5083	33.4791
	gray21.512	33.4435	33.4919	33.3930	33.4781	33.4314	33.4331
	numbers.512	33.4743	33.4759	33.3993	33.4772	33.3567	33.5396
1024 × 1024	ruler.512	33.4256	33.4539	33.5129	33.3883	33.3984	33.4363
	5.3.01	33.4611	33.3901	33.4532	33.4683	33.4741	33.4886
	5.3.02	33.4760	33.3851	33.4853	33.4428	33.4393	33.4384
	7.2.01	33.4289	33.5356	33.4965	33.4688	33.4548	33.4192
	testpat.1k	33.4755	33.4425	33.4455	33.4616	33.4447	33.4452
	Average	33.4640 ¹	33.4560	33.4796	33.4585	33.4597	33.4504
	Std. Dev.	0.02312 ¹	0.04684	0.05158	0.04102	0.04602	0.05610

¹ The bolded values emphasize that IEA-IF-DLT has the best performance in terms of average and stability.

4.5. Histogram Analysis

Natural images usually have significant pixel distribution characteristics, and attackers also hope to find similar distribution characteristics from cipher images. In order to prevent such attacks, an image encryption algorithm must completely eliminate these characteristics, thus preventing attackers from deducing useful information. Figure 11 presents the histograms of some plain images and the corresponding cipher images generated by IEA-IF-DLT. These histograms indicate that the pixel distribution of the plain images is extremely uneven, whereas in the cipher images, these salient features are no longer present. Therefore, IEA-IF-DLT can effectively resist the attacks based on pixel distribution characteristics.

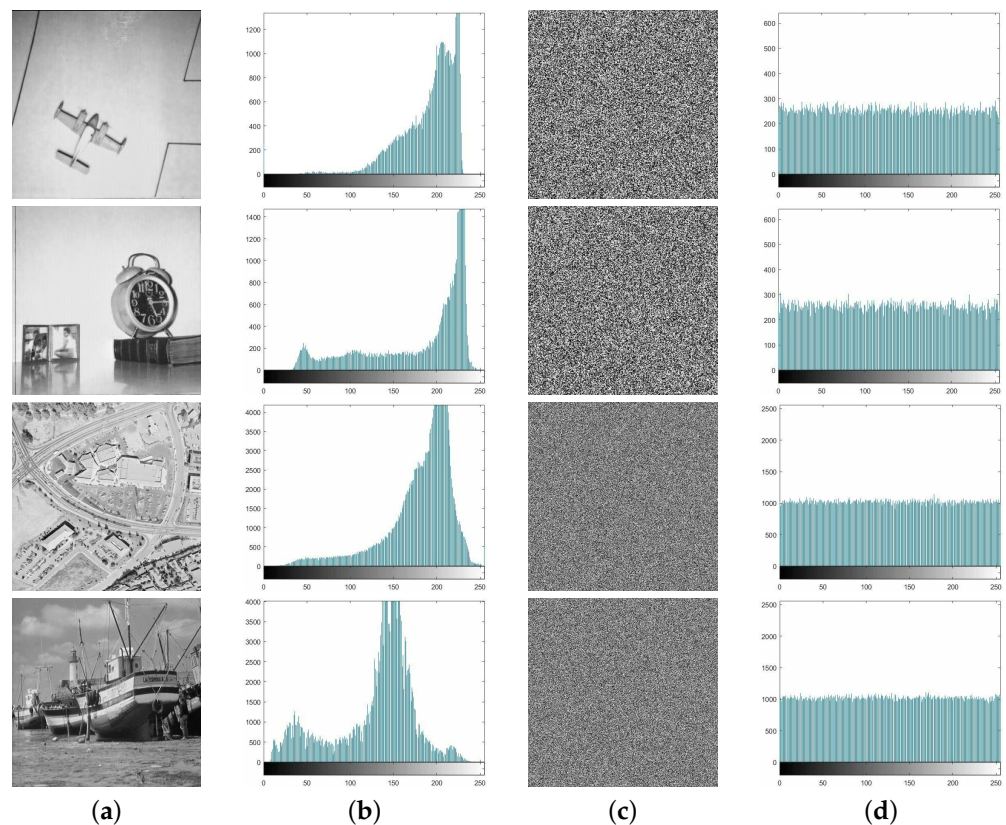


Figure 11. Histograms of plain images and corresponding cipher images: (a) plain images; (b) histograms of plain images; (c) cipher images; (d) histograms of cipher images.

4.6. Correlation Analysis

Adjacent pixels in natural images usually have an extremely high correlation, as shown in the first row of Figure 12. Therefore, a secure image encryption algorithm should be able to effectively eliminate this correlation. The second row of Figure 12 presents the encryption effect of our proposed IEA-IF-DLT. Obviously, the strong correlation of the plain image in the horizontal, vertical, and diagonal directions has been completely eliminated.

Additionally, we introduced the correlation coefficient (CC) to do a quantitative study, so as to more precisely assess how well IEA-IF-DLT performs in reducing the correlation between adjacent pixels. The following is a mathematical definition of CC.

$$CC = \frac{E((v_x - E(v_x)) \times (v_y - E(v_y)))}{\sqrt{D(v_x) \times D(v_y)}}, \tag{22}$$

where $E(v)$ and $D(v)$ represent the expectation and variance of the pixel value v , and v_x, v_y stand for the pixel values of two adjacent pixels in a specific direction. After repeated

calculations, we found that plain images usually have extremely high CC values (>0.8), as demonstrated in Table 5. In contrast, in the cipher images generated by our proposed IEA-IF-DLT, the CC values are extremely low in any direction (<0.005). This further proves that IEA-IF-DLT does have an excellent performance in removing the strong correlation of adjacent pixels in natural images.

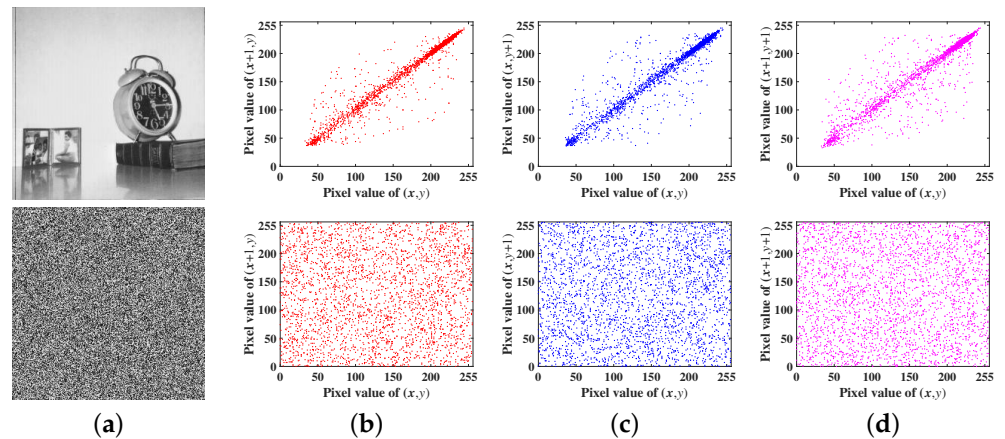


Figure 12. Correlation between adjacent pixels in plain and cipher images: (a) 5.1.12 and its cipher image; (b) horizontal direction; (c) vertical direction; (d) diagonal direction.

Table 5. CCs of different plain and cipher images.

Image Size	Image Type	Filename	CC		
			Horizontal	Vertical	Diagonal
256 × 256	Plain image	5.1.09	0.9389	0.9023	0.9035
		5.1.10	0.8606	0.9051	0.8217
		5.1.11	0.9368	0.9574	0.8921
	Cipher image	5.1.09	0.0014	−0.0007	−0.0019
		5.1.10	−0.0014	−0.0008	−0.0038
		5.1.11	0.0034	0.0026	−0.0046
512 × 512	Plain image	5.2.08	0.8912	0.9366	0.8580
		5.2.09	0.8606	0.9008	0.8028
		5.2.10	0.9279	0.9401	0.8972
	Cipher image	5.2.08	0.0019	−0.0025	−0.0051
		5.2.09	−0.0005	0.0018	0.0022
		5.2.10	0.0010	−0.0017	−0.0039
1024 × 1024	Plain image	5.3.01	0.9812	0.9775	0.9669
		5.3.02	0.9032	0.9104	0.8590
		7.2.01	0.9467	0.9646	0.9448
	Cipher image	5.3.01	0.0012	−0.0020	−0.0004
		5.3.02	0.0018	0.0004	−0.0004
		7.2.01	0.0006	0.0012	0.0009

4.7. Information Entropy Analysis

Information entropy is an indicator that can measure the randomness and distribution uniformity of a signal well, and is widely used to evaluate the security of an encryption algorithm. In a mathematical sense, information entropy can be defined as

$$H(\zeta) = - \sum_{i=1}^T \delta(\zeta_i) \log_2 \delta(\zeta_i), \tag{23}$$

where ζ_i represents one of the symbols whose total number is T , and $\delta(\zeta_i)$ indicates the probability of ζ_i . In general, a higher information entropy value signifies that the signal is

more random and evenly distributed. For an image whose representation depth is eight bits, the ideal information entropy value is 8. We conducted the information entropy test with 20 test images, as shown in Table 6. The information entropy values of the cipher images generated by IEA-IF-DLT are very close to the ideal information entropy value, indicating that they have extremely high randomness. In addition, we also compared the information entropy test results of IEA-IF-DLT with other encryption algorithms, and Table 7 provides the relevant test results. Compared with eight encryption algorithms, the cipher images generated by IEA-IF-DLT have the highest information entropy value, thus demonstrating the excellent performance of IEA-IF-DLT.

Table 6. Information entropy values of commonly used test images and corresponding cipher images generated by IEA-IF-DLT.

Image Size	Image Filename	Information Entropy Value	
		Plain Image	Cipher Image
512 × 512	5.2.08	7.2010	7.9992
	5.2.09	6.9940	7.9992
	5.2.10	5.7056	7.9993
	7.1.01	6.0274	7.9994
	7.1.02	4.0045	7.9992
	7.1.03	5.4957	7.9992
	7.1.04	6.1074	7.9993
	7.1.05	6.5632	7.9994
	7.1.06	6.6953	7.9994
	7.1.07	5.9916	7.9993
	7.1.08	5.0534	7.9993
	boat.512	7.1914	7.9993
	elaine.512	7.5060	7.9992
	gray21.512	4.3923	7.9994
	numbers.512	7.7292	7.9994
	ruler.512	0.5000	7.9992
1024 × 1024	5.3.01	7.5237	7.9998
	5.3.02	6.8303	7.9998
	7.2.01	5.6415	7.9998
	testpat.1k	4.4077	7.9998

Table 7. Information entropy values of Lena cipher images generated by different image encryption algorithms.

Encryption Algorithm	Information Entropy Value
Ref. [47]	7.9992
Ref. [51]	7.9971
Ref. [25]	7.9980
Ref. [52]	7.9909
Ref. [48]	7.9992
Ref. [40]	7.9992
Ref. [43]	7.9992
Ref. [26]	7.9976
IEA-IF-DLT ¹	7.9993 ¹

¹ The bolded value emphasizes that the cipher image generated by IEA-IF-DLT has the highest information entropy value.

4.8. Robustness Analysis

During storage or transmission, images may lose data or be contaminated by noise. In fact, attackers sometimes may also deliberately perform similar attacks on cipher images. Therefore, a robust image encryption algorithm must be able to resist such attacks effectively. To test the robustness of IEA-IF-DLT, that is, its ability to withstand data loss or noise

contamination, we deliberately performed special processing on the House cipher image generated by IEA-IF-DLT. Firstly, we obtained four cipher images by adding different intensities of salt and pepper noise (SPN) to the original cipher image. Then, four additional cipher images were obtained by removing 384×384 pixels at different areas of the red plane. Finally, we decrypted these cipher images. The relevant test results are shown in Figure 13. As one can see, when the cipher image is contaminated by noise, IEA-IF-DLT can still reconstruct the image with high quality, so that almost all the visual information of the plain image can be effectively communicated. Not only that, even if up to 384×384 cipher pixels are missing in different areas, the decrypted images still maintain a very high level of visual quality, thus demonstrating the excellent ability of IEA-IF-DLT to resist such attacks.

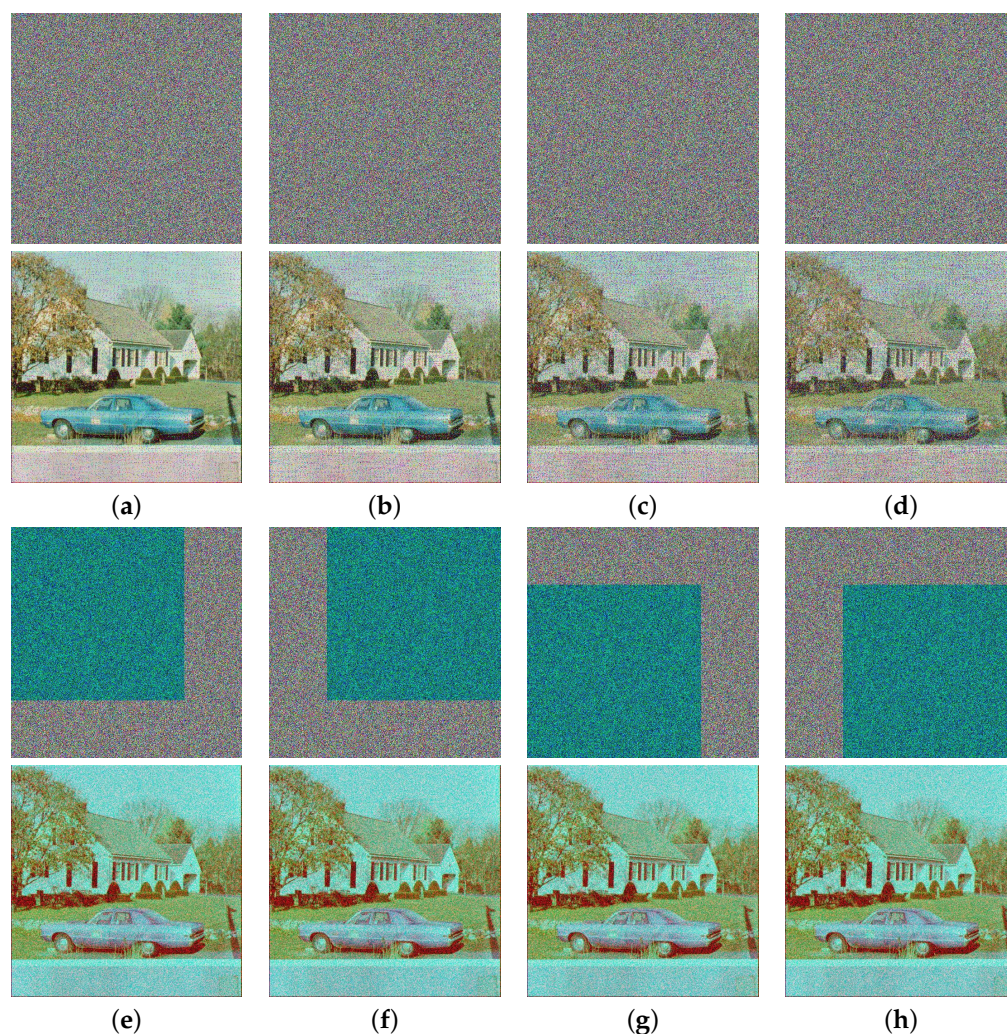


Figure 13. Test results of IEA-IF-DLT in terms of robustness analysis. The first row presents four noise contaminated cipher images, and the second row presents the corresponding decrypted images. The third row shows four cipher images with data loss, and the fourth row shows the corresponding decrypted images. (a) intensity of SPN is 0.005; (b) intensity of SPN is 0.01; (c) intensity of SPN is 0.015; (d) intensity of SPN is 0.02; (e) data loss at top-left corner; (f) data loss at top-right corner; (g) data loss at bottom-left corner; (h) data loss at bottom-right corner.

4.9. Efficiency Analysis

As we all know, in addition to security, improving the efficiency of image encryption is another most important motivation for researchers to design new image encryption algorithms. Therefore, a well-designed image encryption algorithm should not only ensure extremely high security, but also have extremely high encryption efficiency. In our proposed

IEA-IF-DLT, many measures are taken to improve encryption efficiency. Firstly, the way to use the hash value of the plain image is improved. Thus, it is no longer necessary to repeatedly generate chaotic sequences. In other words, once the key is determined, the chaotic sequences can be generated in advance before encryption. Secondly, many high-dimensional operations are employed. Compared with the pixel-by-pixel processing method, the one-dimensional vector-level, two-dimensional plane-level, and especially the three-dimensional operations adopted by IEA-IF-DLT can significantly improve the encryption efficiency. Table 8 provides the times required by IEA-IF-DLT to encrypt test images of four common sizes. It can be seen that IEA-IF-DLT outperforms several state-of-the-art algorithms in terms of encryption efficiency.

Table 8. Times (in seconds) required by different encryption algorithms to encrypt images of four common sizes.

Image Size	256 × 256	512 × 512	1024 × 1024
Ref. [47]	0.0538 s	0.2338 s	1.1494 s
Ref. [40]	0.0800 s	0.4842 s	2.2848 s
Ref. [49]	0.9261 s	3.8887 s	19.3147 s
Ref. [50]	0.3243 s	1.6113 s	7.7342 s
Ref. [48]	0.0949 s	0.4010 s	1.9857 s
Ref. [25]	0.2224 s	0.9731 s	3.8377 s
Ref. [42]	0.6347 s	2.4913 s	9.9185 s
Ref. [53]	0.9810 s	3.8539 s	15.4565 s
IEA-IF-DLT	0.0324 s	0.1638 s	0.9118 s

5. Conclusions

In this paper, a novel image encryption algorithm called IEA-IF-DLT is proposed. The proposed algorithm adopts a binary sequence of length 260 bits as the secret key and uses it to generate the initial state values and parameters of the two employed chaotic systems. This not only avoids the problem of constantly changing secret keys when encrypting different images, but also avoids the need to regenerate chaotic sequences. After transforming the plain image into a 3D image, IEA-IF-DLT utilizes the chaotic sequences to perform three rounds of plane-level permutation, plane-level image filtering, and 3D chaotic image superposition. When compared to bit-level or pixel-by-pixel encryption operations, these encryption operations can significantly improve encryption efficiency. Next, IEA-IF-DLT performs a discrete logarithmic transformation on the intermediate cipher image to ensure that it can effectively resist common differential attacks. Finally, for common plaintext attack strategies, random pixel swapping is incorporated to avoid the diffusion operation being isolated. To evaluate the performance and security of IEA-IF-DLT, we systematically tested and analyzed it, and compared the test results with other advanced algorithms. The test and analysis results show that our proposed algorithm does have better security performance and it also exhibits significant efficiency advantages. In the future, we will further optimize this algorithm and apply it to video encryption.

Author Contributions: Conceptualization, W.F. and X.Z.; methodology, X.Z. and J.Z. (Jing Zhang); software, W.F. and J.Z. (Junkun Zhang); validation, W.F., J.Z. (Jing Zhang) and Z.Q.; formal analysis, W.F.; writing—original draft preparation, W.F. and X.Z.; writing—review and editing, W.F., X.Z. and Z.Q.; supervision, X.Z. and Y.H.; project administration, X.Z. and Y.H.; funding acquisition, W.F., J.Z. (Jing Zhang) and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant Nos. 51977153 and 51637004), the Science and Technology Development Center Project of Chinese Ministry of Education (Grant No. 2021KSA01008), the Project of the Sichuan Higher Education Society of China (Grant No. GJXHXXH21-YB-27), and the Guiding Science and Technology Plan Project of Panzhihua City (Grant No. 2020ZD-S-40).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Ghadirli, H.M.; Nodehi, A.; Enayatifar, R. An overview of encryption algorithms in color images. *Signal Process.* **2019**, *164*, 163–185. [[CrossRef](#)]
2. Zhang, Y.; He, Q.; Xiang, Y.; Zhang, L.Y.; Liu, B.; Chen, J.; Xie, Y. Low-Cost and Confidentiality-Preserving Data Acquisition for Internet of Multimedia Things. *IEEE Internet Things J.* **2018**, *5*, 3442–3451. [[CrossRef](#)]
3. Li, C.; Zhang, Y.; Xie, E.Y. When an attacker meets a cipher-image in 2018: A Year in Review. *J. Inf. Secur. Appl.* **2019**, *48*, 102361. [[CrossRef](#)]
4. Zhang, Y.; He, Q.; Chen, G.; Zhang, X.; Xiang, Y. A Low-Overhead, Confidentiality-Assured, and Authenticated Data Acquisition Framework for IoT. *IEEE Trans. Ind. Inform.* **2020**, *16*, 7566–7578. [[CrossRef](#)]
5. Özkaynak, F. Brief review on application of nonlinear dynamics in image encryption. *Nonlinear Dyn.* **2018**, *92*, 305–313. [[CrossRef](#)]
6. Zhang, Y.; Wang, P.; Huang, H.; Zhu, Y.; Xiao, D.; Xiang, Y. Privacy-Assured FogCS: Chaotic Compressive Sensing for Secure Industrial Big Image Data Processing in Fog Computing. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3401–3411. [[CrossRef](#)]
7. Oravec, J.; Ovsenik, L.; Papaj, J. An Image Encryption Algorithm Using Logistic Map with Plaintext-Related Parameter Values. *Entropy* **2021**, *23*, 1373. [[CrossRef](#)]
8. Zhang, Y.; Wang, P.; Fang, L.; He, X.; Chen, B. Secure Transmission of Compressed Sampling Data Using Edge Clouds. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6641–6651. [[CrossRef](#)]
9. Ahmad, M.; Doja, M.N.; Beg, M.M.S. Security analysis and enhancements of an image cryptosystem based on hyperchaotic system. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 77–85. [[CrossRef](#)]
10. El-Latif, A.A.A.; Abd-El-Atty, B.; Belazi, A.; Iliyasu, A.M. Efficient Chaos-Based Substitution-Box and Its Application to Image Encryption. *Electronics* **2021**, *10*, 1392. [[CrossRef](#)]
11. El-Latif, A.A.A.; Abd-El-Atty, B.; Mazurczyk, W.; Fung, C.; Venegas-Andraca, S.E. Secure Data Encryption Based on Quantum Walks for 5G Internet of Things Scenario. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 118–131. [[CrossRef](#)]
12. Wang, X.; Xue, W.; An, J. Image encryption algorithm based on tent-dynamics coupled map lattices and diffusion of household. *Chaos Solitons Fractals* **2020**, *141*, 110309. [[CrossRef](#)]
13. Pourasad, Y.; Ranjbarzadeh, R.; Mardani, A. A new algorithm for digital image encryption based on chaos theory. *Entropy* **2021**, *23*, 341. [[CrossRef](#)] [[PubMed](#)]
14. Feng, W.; He, Y.; Li, H.; Li, C. A Plain-Image-Related Chaotic Image Encryption Algorithm Based on DNA Sequence Operation and Discrete Logarithm. *IEEE Access* **2019**, *7*, 181589–181609. [[CrossRef](#)]
15. Li, H.; Li, T.; Feng, W.; Zhang, J.; Zhang, J.; Gan, L.; Li, C. A novel image encryption scheme based on non-adjacent parallelable permutation and dynamic DNA-level two-way diffusion. *J. Inf. Secur. Appl.* **2021**, *61*, 102844. [[CrossRef](#)]
16. Zhang, Q.; Guo, L.; Wei, X. A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Opt.-Int. J. Light Electron Opt.* **2013**, *124*, 3596–3600. [[CrossRef](#)]
17. Abd EL-Latif, A.A.; Abd-El-Atty, B.; Abou-Nassar, E.M.; Venegas-Andraca, S.E. Controlled alternate quantum walks based privacy preserving healthcare images in Internet of Things. *Opt. Laser Technol.* **2020**, *124*, 105942. [[CrossRef](#)]
18. Abd-El-Atty, B.; Iliyasu, A.M.; Alanezi, A.; Abd El-latif, A.A. Optical image encryption based on quantum walks. *Opt. Lasers Eng.* **2021**, *138*, 106403. [[CrossRef](#)]
19. Ye, G.; Pan, C.; Dong, Y.; Shi, Y.; Huang, X. Image encryption and hiding algorithm based on compressive sensing and random numbers insertion. *Signal Process.* **2020**, *172*, 107563. [[CrossRef](#)]
20. Chai, X.; Wu, H.; Gan, Z.; Han, D.; Zhang, Y.; Chen, Y. An efficient approach for encrypting double color images into a visually meaningful cipher image using 2D compressive sensing. *Inf. Sci.* **2021**, *556*, 305–340. [[CrossRef](#)]
21. Wang, X.; Liu, C.; Jiang, D. A novel triple-image encryption and hiding algorithm based on chaos, compressive sensing and 3D DCT. *Inf. Sci.* **2021**, *574*, 505–527. [[CrossRef](#)]
22. Hua, Z.; Zhu, Z.; Yi, S.; Zhang, Z.; Huang, H. Cross-plane colour image encryption using a two-dimensional logistic tent modular map. *Inf. Sci.* **2021**, *546*, 1063–1083. [[CrossRef](#)]
23. Wang, X.; Chen, X. An image encryption algorithm based on dynamic row scrambling and Zigzag transformation. *Chaos Solitons Fractals* **2021**, *147*, 110962. [[CrossRef](#)]
24. Bezerra, J.I.M.; de Almeida Camargo, V.V.; Molter, A. A new efficient permutation-diffusion encryption algorithm based on a chaotic map. *Chaos Solitons Fractals* **2021**, *151*, 111235. [[CrossRef](#)]
25. Diaconu, A.V. Circular inter-intra pixels bit-level permutation and chaos-based image encryption. *Inf. Sci.* **2016**, *355–356*, 314–327. [[CrossRef](#)]
26. Zefreh, E.Z. An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions. *Multimed. Tools Appl.* **2020**, *79*, 24993–25022. [[CrossRef](#)]
27. Feng, W.; He, Y. Cryptanalysis and improvement of the hyper-chaotic image encryption scheme based on DNA encoding and scrambling. *IEEE Photonics J.* **2018**, *10*, 7909215. [[CrossRef](#)]

28. Feng, W.; He, Y.; Li, H.; Li, C. Cryptanalysis and improvement of the image encryption scheme based on 2D logistic-adjusted-sine map. *IEEE Access* **2019**, *7*, 12584–12597. [[CrossRef](#)]
29. Feng, W.; He, Y.; Li, H.; Li, C. Cryptanalysis of the integrated chaotic systems based image encryption algorithm. *Optik* **2019**, *186*, 449–457. [[CrossRef](#)]
30. Feng, W.; Zhang, J. Cryptanalyzing a Novel Hyper-Chaotic Image Encryption Scheme Based on Pixel-Level Filtering and DNA-Level Diffusion. *IEEE Access* **2020**, *8*, 209471–209482. [[CrossRef](#)]
31. Feng, W.; Qin, Z.; Zhang, J.; Ahmad, M. Cryptanalysis and Improvement of the Image Encryption Scheme Based on Feistel Network and Dynamic DNA Encoding. *IEEE Access* **2021**, *9*, 145459–145470. [[CrossRef](#)]
32. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [[CrossRef](#)]
33. Preishuber, M.; Hütter, T.; Katzenbeisser, S.; Uhl, A. Depreciating motivation and empirical security analysis of chaos-based image and video encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2137–2150. [[CrossRef](#)]
34. Gao, T.; Chen, Z.; Yuan, Z.; Yu, D. Adaptive synchronization of a new hyperchaotic system with uncertain parameters. *Chaos Solitons Fractals* **2007**, *33*, 922–928. [[CrossRef](#)]
35. Li, C.; Feng, B.; Li, S.; Kurths, J.; Chen, G. Dynamic analysis of digital chaotic maps via state-mapping networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2019**, *66*, 2322–2335. [[CrossRef](#)]
36. Li, C.; Tan, K.; Feng, B.; Lü, J. The graph structure of the generalized discrete Arnold's Cat map. *IEEE Trans. Comput.* **2022**, *71*, 364–377. [[CrossRef](#)]
37. Sahasrabudhe, A.; Laiphrakpam, D.S. Multiple images encryption based on 3D scrambling and hyper-chaotic system. *Inf. Sci.* **2021**, *550*, 252–267. [[CrossRef](#)]
38. Gan, Z.; Chai, X.; Zhi, X.; Ding, W.; Lu, Y.; Wu, X. Image cipher using image filtering with 3D DNA-based confusion and diffusion strategy. *Neural Comput. Appl.* **2021**, *33*, 16251–16277. [[CrossRef](#)]
39. Feng, W.; He, Y.; Li, H.; Li, C. Image encryption algorithm based on discrete logarithm and memristive chaotic system. *Eur. Phys. J. Spec. Top.* **2019**, *228*, 1951–1967. [[CrossRef](#)]
40. Hua, Z.; Jin, F.; Xu, B.; Huang, H. 2D Logistic-Sine-coupling map for image encryption. *Signal Process.* **2018**, *149*, 148–161. [[CrossRef](#)]
41. Feng, W.; Zhang, J.; Qin, Z. A Secure and Efficient Image Transmission Scheme Based on Two Chaotic Maps. *Complexity* **2021**, *2021*, 1898998. [[CrossRef](#)]
42. Hua, Z.; Zhou, Y. Design of image cipher using block-based scrambling and image filtering. *Inf. Sci.* **2017**, *396*, 97–113. [[CrossRef](#)]
43. Hua, Z.; Xu, B.; Jin, F.; Huang, H. Image Encryption Using Josephus Problem and Filtering Diffusion. *IEEE Access* **2019**, *7*, 8660–8674. [[CrossRef](#)]
44. Ma, Y.; Li, C.; Ou, B. Cryptanalysis of an Image Block Encryption Algorithm Based on Chaotic Maps. *J. Inf. Secur. Appl.* **2020**, *54*, 102566. [[CrossRef](#)]
45. Liu, S.; Li, C.; Hu, Q. Cryptanalyzing two image encryption algorithms based on a first-order time-delay system. *IEEE Multimed.* **2022**, *29*, 74–84. [[CrossRef](#)]
46. Chen, L.; Li, C.; Li, C. Security Measurement of a Medical Image Communication Scheme based on Chaos and DNA. *J. Vis. Commun. Image Represent.* **2022**, *83*, 103424. [[CrossRef](#)]
47. Hua, Z.; Zhou, Y.; Pun, C.M.; Chen, C.P. 2D Sine Logistic modulation map for image encryption. *Inf. Sci.* **2015**, *297*, 80–94. [[CrossRef](#)]
48. Hua, Z.; Zhou, Y.; Huang, H. Cosine-transform-based chaotic system for image encryption. *Inf. Sci.* **2019**, *480*, 403–419. [[CrossRef](#)]
49. Zhu, H.; Zhao, Y.; Song, Y. 2D Logistic-Modulated-Sine-Coupling-Logistic Chaotic Map for Image Encryption. *IEEE Access* **2019**, *7*, 14081–14098. [[CrossRef](#)]
50. Cao, C.; Sun, K.; Liu, W. A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. *Signal Process.* **2018**, *143*, 122–133. [[CrossRef](#)]
51. Wang, X.; Liu, L.; Zhang, Y. A novel chaotic block image encryption algorithm based on dynamic random growth technique. *Opt. Lasers Eng.* **2015**, *66*, 10–18. [[CrossRef](#)]
52. Wu, X.; Wang, D.; Kurths, J.; Kan, H. A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system. *Inf. Sci.* **2016**, *349–350*, 137–153. [[CrossRef](#)]
53. 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](#)]