fractal and fractional

MDPI

Article

Fast Encryption Algorithm Based on Chaotic System and Cyclic Shift in Integer Wavelet Domain

Yuan-Min Li *, Yang Deng, Mingjie Jiang and Deyun Wei

School of Mathematics and Statistics, Xidian University, Xi'an 710071, China; dy3581908802@163.com (Y.D.); dywei@xidian.edu.cn (D.W.)
* Correspondence: ymli@xidian.edu.cn

Abstract: This paper introduces a new fast image encryption scheme based on a chaotic system and cyclic shift in the integer wavelet domain. In order to increase the effectiveness and security of encryption, we propose a new diffusion scheme by using bidirectional diffusion and cyclic shift and apply it to our encryption scheme. First, a two-level integer wavelet transform is used to split the plaintext picture into four low-frequency components. Second, we use random sequences generated by Chen's hyper-chaotic system to scramble four low-frequency components. The initial value is determined by Secure Hash Algorithm 256-bit (SHA256) and user-defined parameters, which increases the plaintext sensitivity. Then, the new diffusion scheme is applied to the matrix containing most of the information and matrices are transformed by a one-level inverse integer wavelet. Finally, to create the ciphertext image, the diffused matrices are subjected to the one-level inverse integer wavelet transform. In the simulation part, we examine the suggested algorithm's encryption impact. The findings demonstrate that the suggested technique has a sufficient key space and can successfully fend off common attacks.

Keywords: image encryption; integer wavelet transform; Chen's hyper-chaotic map; cyclic shift

Citation: Li, Y.-M.; Deng, Y.; Jiang, M.; Wei, D. Fast Encryption Algorithm Based on Chaotic System and Cyclic Shift in Integer Wavelet Domain. *Fractal Fract.* **2024**, *8*, 75. https://doi.org/10.3390/fractalfract8020075

Academic Editors: Ahmed I. Zayed, Bingzhao Li, Zunwei Fu and Xiangyang Lu

Received: 8 November 2023
Revised: 12 January 2024
Accepted: 22 January 2024
Published: 24 January 2024

1. Introduction

Information security is becoming increasingly crucial as science and technology advance. The image encryption algorithm is an important technical means to protect user privacy and image security. Many scholars have adopted various encryption techniques in encryption algorithms to ensure image security [1–11]. Common encryption techniques are optical transform [1–3], DNA operation [4,5,7,8], chaotic system [9,10], compressed sensing [11], and so on. The great sensitivity, unpredictable nature, and ergodicity of these systems to beginning conditions means that chaotic systems are the most commonly used of these technologies [1,9,11].

Fridrich suggested a conventional encryption technique based on a chaotic map in 1998 [12]. It involves two diffusion and permutation processes that must be run numerous times. A novel logistic map-based picture encryption technique was proposed by Munoz and colleagues [13]. The deformed chaotic map can describe a parameter region for selecting the key, thus expanding the key space and avoiding the nonchaotic region. An encryption scheme has been presented using a spatiotemporal chaotic system [14]. However, the above encryption algorithm is broken because it is vulnerable to plaintext attacks [15]. Later, high-dimensional chaos was proposed and was applied more and more to encryption algorithms. High-dimensional chaos has more complicated dynamic behavior than low-dimensional chaos, as well as greater starting value sensitivity and improved protection [16–19]. For example, a scale-invariant digital photo encryption approach based on a 3D modular chaotic map with high levels of security capacity and effectiveness was proposed by Hamed et al. [20]. A hyper-chaotic picture cryptography was proposed utilizing a pixel-permutation and bit-confusion approach, which can endure typical attacks and overcome

the disadvantages of low-dimensional chaotic map methods [17]. To sum up, the hyper-chaotic system has many good properties, which makes the hyper-chaotic system very suitable for image encryption [21].

Although the difficulty of the frequency domain computation is greater than the spatial encryption scheme, it often has a better encryption effect [22–24]. Wavelet transform is commonly used in frequency domain encryption algorithms because it can decompose the low-frequency matrix containing the most information. Processing low-frequency information can considerably increase the effectiveness of encryption [25–28]. Based on DWT and QR decomposition technologies, Rakheja et al. devised an asymmetric picture encryption technique. Through wavelet transform and matrix decomposition technology, the small-size matrix contains most of the information so as to decrease the time required for encryption [29]. A method for multiple-image encryption using the integer wavelet transform (IWT) is presented [24]. The outcomes of the simulation demonstrate that IWT has superior qualities to DWT. Refs. [30,31] proposes an adaptive chaotic, wavelet, and cyclic algorithm for digital picture encryption.

The results of the research indicated above demonstrate that the employment of wavelet technology considerably improves the efficiency of the encryption system. However, some common encryption stages, such as DNA coding and diffusion operations, are often not improved. To ensure the effectiveness of encryption and enhance the algorithm's security, cyclic shift is introduced into the design of an encryption algorithm [30–33]. Because of the simple operation of cyclic shift, the combination with other encryption techniques can ensure good encryption results with high efficiency. Initially, cyclic shift is often combined with bit-plane encryption algorithms [32,33]. The encryption process is used by expanding the bit plane of the image to carry out cyclic shift. Later, Wang et al. applied cyclic shift to the DNA coding stage, which considerably enhanced the algorithm's security and encryption efficiency [30,32].

Using integer wavelet transform and cyclic shift, we suggest a quick encryption algorithm in this paper. The four low-frequency sections of the plaintext image are jumbled by random sequences produced by the Chen hyper-chaotic system using a two-level integer wavelet. So as to make sure that the encryption's efficiency is high, we introduce cyclic shift operation in the bidirectional diffusion stage and apply this diffusion method to multiple matrices after integer wavelet decomposition. The approach is highly resistant to different plaintext attacks because the random sequence used in the scrambling step is created by the Chen hyper-chaotic system, and the user-defined and plaintext image's Secure Hash Algorithm 256-bit (SHA256) values make up the system key. The key is created using a piecewise linear chaotic map (PWLCM) during the diffusion phase. Inverse integer wavelet transformation can be used to obtain corresponding ciphertext images from the image after scrambling and diffusion.

The following are the primary contributions of the proposed algorithm:

(1) The encryption algorithm is proposed based on integer wavelet transform, chaotic system, and cyclic shift. The use of integer wavelet transform and cyclic shift not only guarantees the effect of encryption, but also improves the efficiency of encryption.
(2) The hyper-chaotic systems with high unpredictability are used for image scrambling and diffusion, which make the algorithm resistant to various types of statistical attacks.
(3) The initial value of the hyper-chaotic system is generated by SHA256 and user definition, which greatly expands the key space and makes the algorithm very sensitive to the key.
(4) Experimental results show that the proposed algorithm can resist statistical attacks and differential attacks well, and has a large enough key space and strong key sensitivity and security.

This essay's remaining sections are organized as follows. Fundamental notions are presented in Section 2. The suggested image encryption technique is further explained in Section 3. Section 4 provides some simulations and conversations to illustrate performance and security. In Section 5, a succinct conclusion is provided.

2. Background Theory

Here is a brief introduction to some of the mathematical theories incorporated into the suggested encryption scheme.

2.1. Integer Wavelet Transform

During the encryption and decryption procedure, reconstructed images after traditional wavelet transform may be distorted, especially after multilevel wavelet decomposition, which will cause a more serious quality degradation of decrypted images. The significance of integer wavelet transform is that its wavelet coefficients are all integers, so even after multistage wavelet decomposition there will be no distortion after inverse transformation [34]. Thus, in this paper, we apply integer wavelet transform for image encryption.

Integer wavelet transform is usually realized by a lifting scheme, and its process can be divided into three steps including split, prediction, and update [35]. Taking Haar wavelet base as an example, the concrete steps are explained.

Split is usually to divide the original signal $S_{j,k}$ into an even sequence and an odd sequence.

$$split(g_{j,k}) = (g_{j,2k}, g_{j,2k+1}) = (g_{j+1,k}, c_{j+1,k}) \tag{1}$$

In the above equation, P is the prediction operator, which predicts the odd sequence by even sequence $c_{j+1,k} = P(g_{j+1,k})$. The prediction operator based on Haar wavelet is $P(g_{j+1,k}) = g_{j+1,k}$, so that:

$$c_{j+1,k} = c_{j+1,k} - P(g_{j+1,k}) = c_{j+1,k} - g_{j+1,k} \tag{2}$$

The updating operator of Haar wavelet lifting scheme is [35]:

$$\mathrm{U}(c_{j+1,k}) = \left\lfloor \frac{c_{j+1,k}}{2} \right\rfloor + \left\lfloor \frac{c_{j+1,k+1} - c_{j+1,k}}{8} \right\rfloor \tag{3}$$

where $\lfloor \cdot \rfloor$ is the rounding operation.

Through the above process, it can be seen that the even sequence contains most of the information of the signal, while the odd sequence mainly contains the details of the signal. Figure 1 depicts the two-level wavelet decomposition of a picture. After decomposition, four low-frequency matrices ALL, BLL, CLL, and DLL can be obtained, among which the ALL matrix contains most of the original image's information.

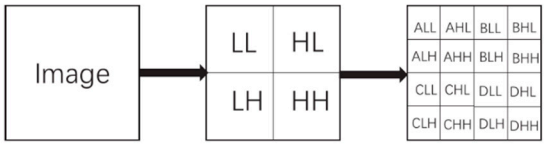

Figure 1. Two-level Wavelet Decomposition.

Two-level integer wavelet transform may result in the loss of information due to the reduced length of the signal. In some cases, this loss of information may be unacceptable, especially for applications that require high-quality and precise restoration. Through the two-level integer wavelet transform of the image, the image can be represented as the approximation and detail coefficients of different frequencies, so it has a wide range of applications in image compression, signal analysis, and data encryption.

2.2. Chen Hyper-Chaotic System

Numerous chaotic systems with good unpredictability have been defined in recent years. In picture encryption, more and more chaotic methods are being used. Based on Chen chaos, Li et al. suggested the Chen hyper-chaotic system [36]. Equation (4) defines the Chen hyper-chaotic system.

$$\dot{r} = a(s - r)$$

$$\begin{aligned} \dot{s} &= dr - rt + cs - h \\ \dot{t} &= rs - bt \\ \dot{u} &= r + h \end{aligned} \tag{4}$$

where the Chen hyper-chaotic system's control parameters are a, b, c, d, and h.

The Lyapunov exponent is a crucial metric for assessing how chaotic maps behave dynamically when the following conditions are met: $a = 36$, $b = 3$, $c = 28$, $d = -16$, and $h \in (-0.7, 0.7)$. The Chen hyper-chaotic system's attractors are depicted in Figure 2. The chaotic system exhibits good ergodicity, as can be demonstrated. The Chen hyper-chaotic system's prediction time is quicker and more sensitive than that of low-dimensional chaotic systems. Therefore, the use of random sequences generated by the Chen hyper-chaotic system in the encryption scheme can improve security.

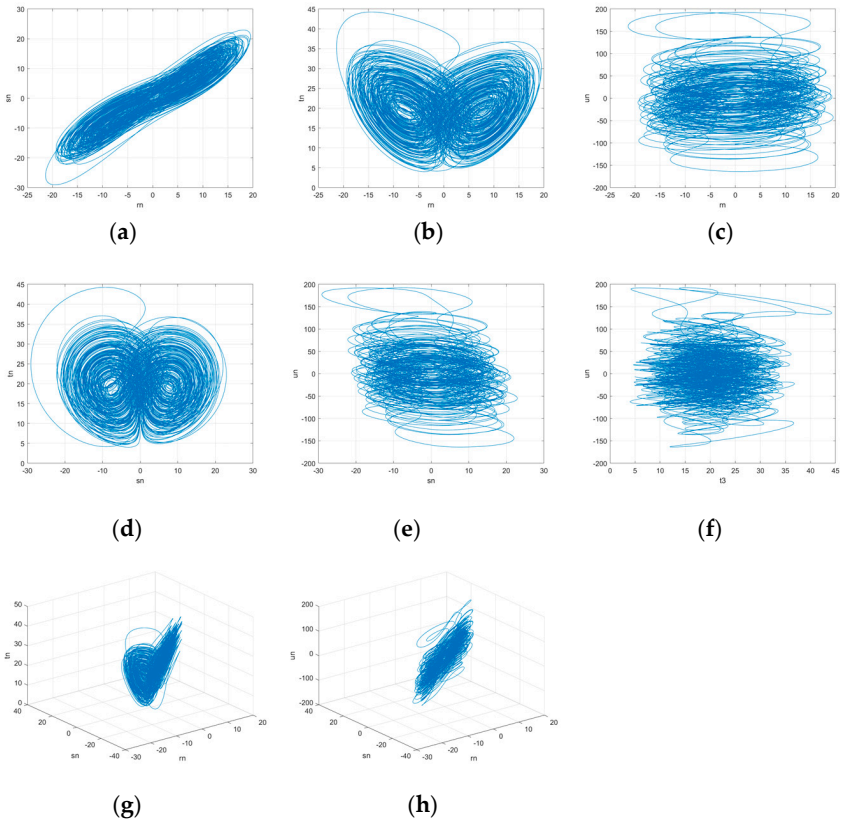

Figure 2. Chen hyper-chaotic attractor. (**a**) (r–s); (**b**) (r–t); (**c**) (r–u); (**d**) (s–t); (**e**) (s–u); (**f**) (t–u); (**g**) (r–s–t); (**h**) (r–s–u).

2.3. Piecewise Linear Chaotic Map (PWLCM)

$$y_i = f(y_{i-1}, \mu) = \begin{cases} y_{i-1}/\mu, 0 \le y_{i-1} \le \mu \\ (y_{i-1} - \mu)/(0.5 - \mu), \mu \le y_{i-1} < 0.5 \\ 0, y_{i-1} = 0.5 \\ f(1 - y_{i-1}, \mu), 0.5 < y_{i-1} < 1.0 \end{cases} \tag{5}$$

where $y_i \in$ (0,1) and $\mu \in$ (0,0.5), which is a parameter. The PWLCM system is well suited to producing random sequences since it has an excellent ergodicity and uniform distribution.

3. Results

The three stages of the algorithm described in this paper are key generation, permutation, and dissemination. In the permutation stage, we primarily use the chaotic sequence produced by Chen's hyper-chaos to confuse the four low-frequency matrices acquired after

second-order integer wavelet decomposition. We suggest a novel bidirectional diffusion strategy based on cyclic displacement for the diffusion stage. PWLCM generates the diffusion sequence during the diffusion step. The A_{LL} matrix, which contains the majority of the plaintext image's information, and the matrix derived from the first-order inverse integer wavelet are both subjected to this technique to increase the algorithm's security. Figure 3 depicts the encryption flowchart. Below, more information about these three stages' specifics will be provided.

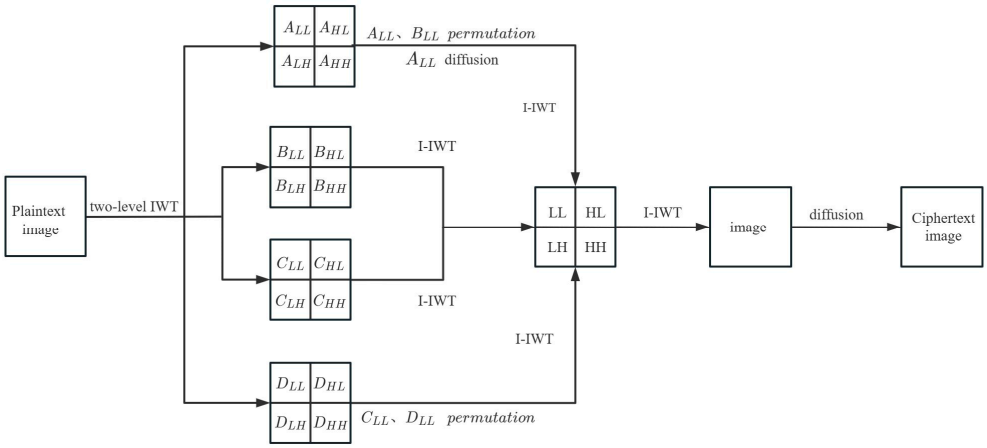

Figure 3. The process of encryption.

3.1. The Generation of Key

The key in both the scrambling phase and the diffusion phase includes two parts. The user of the system supplies one component, while the plaintext image's SHA256 hash value determines the other. The SHA256 hash value is highly dependent on the original information; therefore, if the image is altered a small amount, the related hash result will vary dramatically. As a result, by employing the SHA256 hash value, the method may successfully fend off a plaintext attack.

SHA256 is a hash function in the SHA-2 (Secure Hash Algorithm 2) family that produces a 256-bit (32 byte) hash value, usually in the form of 64 hexadecimal characters. SHA256 is widely used in cryptography, digital signature, data integrity verification, and other fields to provide the secure hashing of data. We put every 8 bits of binary into a block and represent them as $k_1, k_2, k_3 \ldots k_{32}$ ($k^i = \{k_0^i, k_1^i, k_2^i, \ldots, k_7^i\}$). $k_1, k_2, k_3 \ldots, k_{32}$ can be expressed in decimal notation. Next, we use the keys generated by SHA256 and the user-defined keys $x_0', y_0', z_0', w_0', xx_1'$, and yy_1' to create the PWLCM and Chen's hyper-chaotic system's initial values.

The Chen hyper-chaotic system's starting values are as follows:

$$x_0 = mod[(k_1 + k_2 + k_3) \oplus (k_1 + k_2 + k_3), 1] + x_0' \tag{6}$$

$$y_0 = mod[(k_7 + k_8 + k_9) \oplus (k_{10} + k_{11} + k_{12}), 2] + y_0' \tag{7}$$

$$w_0 = mod[(k_{13} + k_{14} + k_{15}) \oplus (k_{16} + k_{17} + k_{18}), 3] + w_0' \tag{8}$$

$$z_0 = mod[(k_{19} + k_{20} + k_{21}) \oplus (k_{22} + k_{23} + k_{24}), 4] + z_0' \tag{9}$$

These are PWLCM's initial values and parameters:

$$\begin{cases} xx_1 = mod\left[\frac{(k_{25}+k_{26})\oplus(k_{27}+k_{28})}{256} + xx_1', 1\right] \\ yy_1 = mod\left[\frac{(k_{29}+k_{30})\oplus(k_{31}+k_{32})}{256} + yy_1', 0.5\right] \end{cases} \tag{10}$$

3.2. The Process of Permutation

These are the precise steps:

Step 1: the initial values produced by Equation (6) are entered into Chen's hyperchaotic system to produce four random sequences of length $MN/16$: X, Y, Z, and W.

Step 2: The coefficients of the four low-frequency matrices are summed separately, and the random sequence for scrambling is determined by the following formula:

$$a = mod(sum, 4) + 1 \tag{11}$$

where sum is the sum of the wavelet coefficients. When a = 1, this low-frequency part uses random sequence X to permutate. When a = 2, this low-frequency part uses random sequence Y to permutate and so on.

Step 3: Assume that current matrix is A_{LL} and the chaotic sequence X is used for scrambling. We round X and map it modulo to integers from 1 to $MN/16$.

Step 4: X is extended by the values from the set $\{1, 2, \cdots, MN/16\}$ that do not occur in X. Repeat numbers in X are reserved only for the first occurrence. Swap the positions of $A_{LL}(X_i)$ and $A_{LL}(X_{MN/16-i+1})$.

3.3. The Process of Diffusion Based on Cyclic Shift

In this section, we introduce traditional diffusion operations and propose a diffusion scheme based on bidirectional diffusion and cyclic shift. The proposed scheme is applied to the scrambled matrices, and the detailed steps are given below. The principle of traditional diffusion based on addition and modulus operation is:

$$C_i = (C_{i-1} + S_i + P_i) \bmod 256 \tag{12}$$

where P_i is the vector that expands from the plaintext image, S_i is random sequence, and C_i is the encryption vector obtained. By circularly expanding Equation (9) we can obtain:

$$C_n = (C_0 + S_1 + \cdots + S_n + P_1 + \cdots + P_n) \bmod 256 \tag{13}$$

It can be obtained from Equation (10) that the information of the plaintext pixel P_i can only be hidden in $C_i \sim C_N$. This is due to the fact that the above operation is forward diffusion. Therefore, if we want to hide the plaintext information P_i in the whole ciphertext sequence, diffusion operation needs to be looped twice: forward diffusion and backward diffusion.

Because cyclic shift operates on binary bit, it has high efficiency. We introduce cyclic shift into the diffusion process. It can significantly increase the algorithm's security. Equation (10) contains the proposed diffusion operation.

$$C_i = (C_{i-1} + M_i + N_i) \bmod 256 <<< \mathrm{LSB}_3\ (\mathrm{C}_{i-1}) \tag{14}$$

where LSB_3 stands for moving the data's three lowest bits. Based on the proposed scheme above, the following steps can be attained.

Step 1: expanded into a one-dimensional vector with rows and columns, the original, two-dimensional picture matrix p is given the name N.

Step 2: using the initial values xx_1 and yy_1 of the PWLCM from Equation (7), we generate two diffusion sequences: S_1 and S_2.

Step 3: S_1 and S_2 are used for forward and backward diffusion of P according to the proposed diffusion operation.

4. Simulation Results and Security Analysis

Here, we simulate the suggested algorithm and examine the outcomes of its encryption and decryption effects. We contrast the suggested method with the related algorithms in each area.

4.1. Encryption and Decryption Results

We use 512 × 512 images including "Man", "Zelda", "Mandrill", "Einstein", "House", and "Building". Figure 4 displays their original photos as well as the accompanying encrypted and decrypted images. The parameter values used by the proposed encryption algorithm are $x'_0 = 1.1$, $y'_0 = 2.2$, $z'_0 = 3.3$, $w'_0 = 4.4$, $xx'_1 = 0.1$, and $yy'_1 = 0.1$.

As observed in Figure 4, the encrypted image resembles noise, demonstrating the effectiveness of the encryption technique in concealing the contents of the plaintext image. Consequently, the suggested algorithm can enact effective encryption and decryption.

4.2. Key Space Analysis

Key space is a significant indicator of how secure an encryption technique is [37,38]. The proposed encryption algorithm's key space must be sufficiently large to withstand strong attacks. In the suggested cryptographic algorithm, there are six parameters in the encryption process, which are x_0, y_0, z_0, and w_0 for Chen's hyper-chaotic system in permutation stage and xx_1 and yy_1 for PWLCM in the diffusion stage. Given a computer with a computing precision of 10^{-14}, the key space is about $\left(10^{14}\right)^6 = 10^{84} \approx 2^{280}$. This shows that the key space of our method is sufficiently large—much larger than that of 2^{100}—to withstand powerful attacks.

4.3. Statistical Analysis

This subsection focuses mostly on how well the algorithm can withstand statistical analysis.

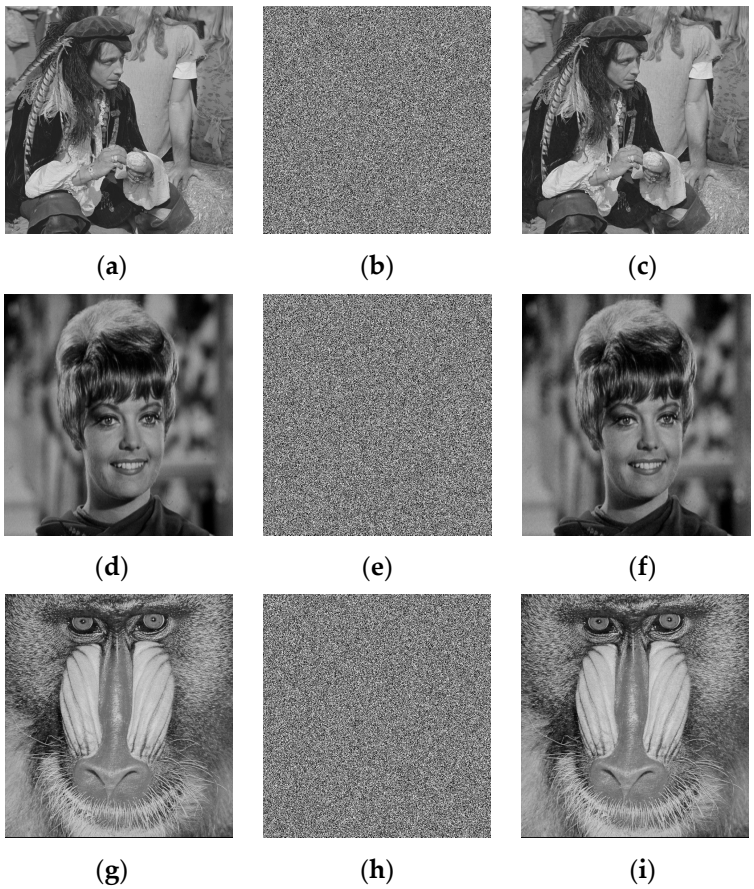

Figure 4. *Cont.*

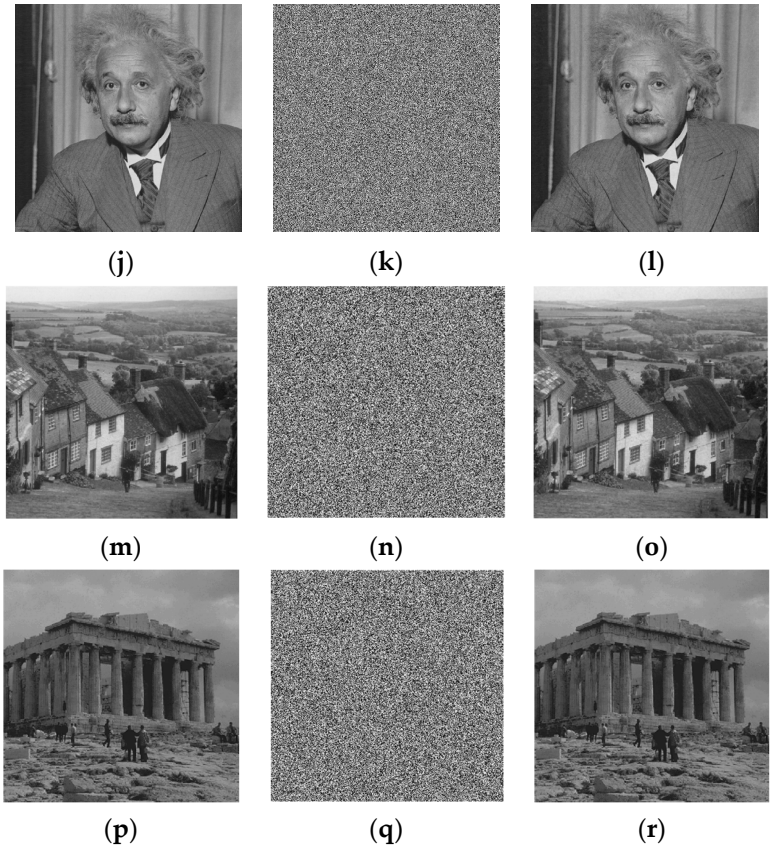

Figure 4. Encryption and decryption simulation results. (**a**,**d**,**g**,**j**,**m**,**p**) Plaintext image; (**b**,**e**,**h**,**k**,**n**,**q**) Ciphertext image; (**c**,**f**,**i**,**l**,**o**,**r**) Decrypt image.

4.3.1. Histogram Analysis

When analyzing a grayscale image, the histogram typically counts the frequency of each pixel between 0 and 225. Plaintext images generally have a specific meaning, so the pixel values are generally concentrated in one or more pixel segments. The encrypted image's pixel distribution needs to be more evenly distributed in order for the encryption technique to withstand statistical attacks. Figure 5 displays the histogram statistics of the photos "Boat", "Flinstones", "Goldhill", and "Lena" both before and after encryption. Histogram statistical findings of encrypted images are evenly distributed, as can be observed from (b), (d), (f), and (h) in Figure 5, whereas those of plaintext images have specific features. This demonstrates how effectively the suggested technique may conceal the image's content and how challenging it would be for an attacker to decrypt the image and obtain relevant information.

4.3.2. Correlation Coefficient Analysis

There is frequently a strong correlation between neighboring pairs of pixels in a plaintext image. This feature will probably be used by the attacker to break the encryption scheme. Through the use of encryption algorithms, the correlation between adjacent pixel pairs must be reduced. Typically, correlation is calculated using the correlation coefficient, which is defined as [39]:

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{\mathrm{D}(x) - \mathrm{D}(y)}} \tag{15}$$

$$\operatorname{cov}(x,y) = E[x - E(x)][y - E(y)] \tag{16}$$

x and y usually refer to the pixel values of horizontal or vertical or diagonal pairs of pixels. The correlation is stronger and smaller, respectively, depending on how near the correlation coefficient's absolute value is to 1.

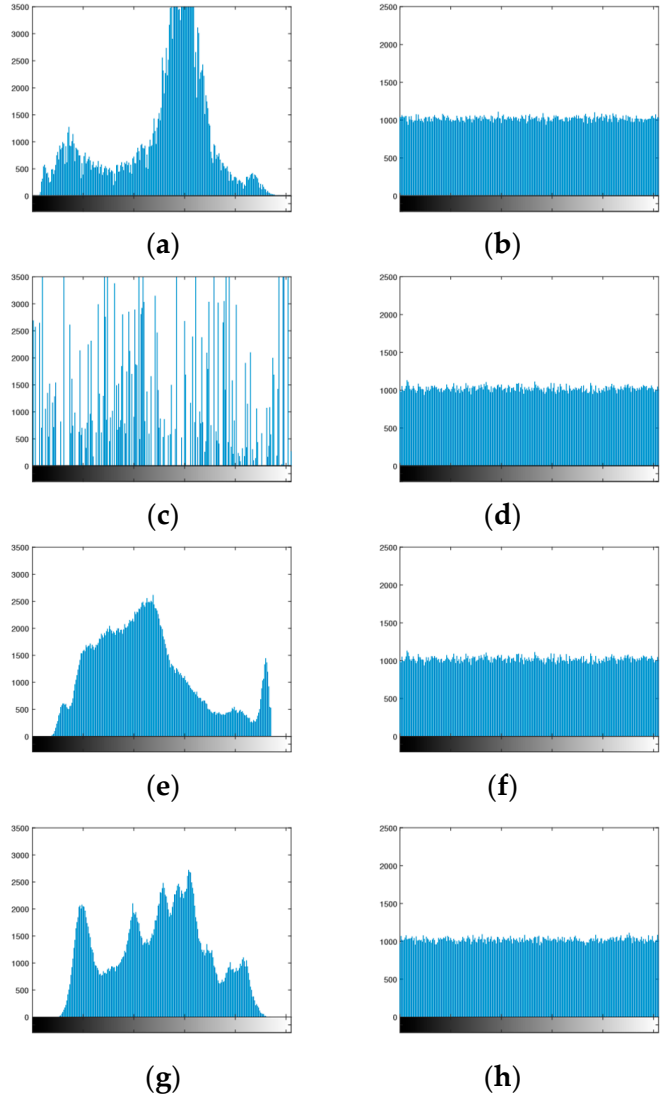

Figure 5. Histogram analysis. Plaintext image: (**a**,**c**,**e**,**g**); Ciphertext image: (**b**,**d**,**f**,**h**).

We choose 5000 adjacent pixel pairs from the plaintext images "Einstein" and "Boat" in three directions. Their pixel values are counted, and the results are displayed in Figure 6. It is evident from Figure 6 that the adjacent pixel pairs before encryption are concentrated in a diagonal direction, demonstrating how closely the pixel values of the subsequent pixel pairs are spaced, while the encrypted neighboring pixel pairs' pixel values are uniformly distributed, indicating that the distribution of pixel values is very random.

We determined the correlation of nearby pixel pairs in 10 images using the same procedure as described above for choosing adjacent pixel pairs. The findings are displayed in Table 1. Table 2 makes it evident that the correlation is relatively small after encryption because it is close to 1 in absolute value before encryption, whereas, after encryption, the correlation coefficient's absolute value is near to 0. There is very little association between adjacent pixels. The ability of various encryption techniques to lessen pixel correlation is also compared in Table 2 [39–41]. Table 1 demonstrates that our approach performs better in lowering the correlation between adjacent pixels. In summary, the suggested technique can withstand statistical attacks effectively.

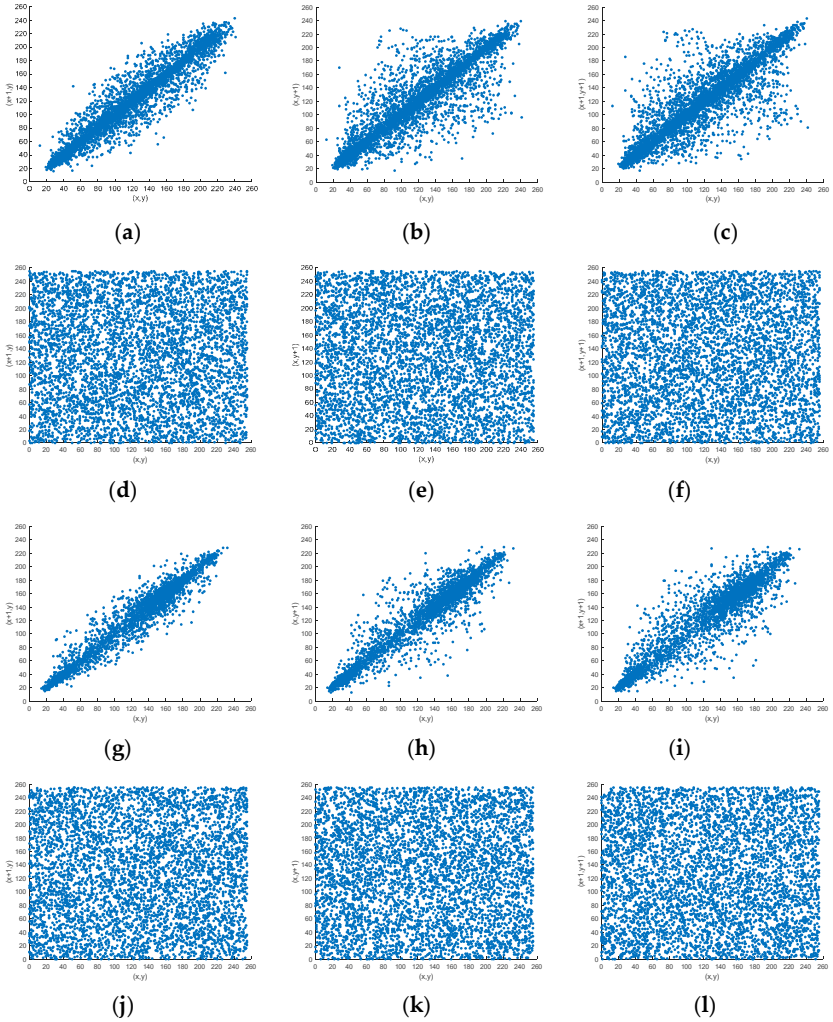

Figure 6. Analysis of correlation coefficients. The Einstein correlations from the three directions are shown in (**a**–**c**), respectively. The equivalent Einstein correlations following encryption are (**g**–**i**). (**d**–**f**) are the correlations of Boat from three directions respectively. (**j**–**l**) are the corresponding correlations of Boat after encryption.

Table 1. Correlation coefficients of different images.

Images	Horizontal	Vertical	Diagonal
Man	0.9592 −0.0001	0.9682 −0.0059	0.9402 0.0302
Zelda	0.9827 0.0107	0.9921 0.0032	0.9788 −0.0149
Einstein	0.9718 −0.0101	0.9804 0.0045	0.9571 −0.0013
Lena	0.9734 0.0171	0.9854 0.0081	0.9611 −0.0119
Mandrill	0.8746 −0.0081	0.7950 0.0042	0.7518 −0.01544
Gold hill	0.9717 0.0310	0.9739 −0.0220	0.9535 0.0060

Table 1. *Cont.*

Images	Horizontal	Vertical	Diagonal
Flintstones	0.9491 0.0139	0.9427 -0.0053	0.9069 −0.0210
Bridge	0.9412 −0.0008	0.9305 0.0063	0.9012 −0.0021
Crowd	0.9068 0.0213	0.9095 0.0090	0.8479 −0.0059
Boat	0.9406 −0.0040	0.9707 0.0209	0.9239 −0.0151

Table 2. Correlation coefficients of different methods.

Direction	Zhou et al. [39]	Zhou et al. [40]	Zhou et al. [41]	Ours
Horizontal	0.0846	0.0198	0.0104	−0.00467
Vertical	0.0583	0.0141	0.0299	0.0135
Diagonal	0.0931	0.0026	0.0062	−0.0055

4.3.3. Information Entropy Analysis

The information entropy of gray image I can be defined as [42]:

$$H(I) = -\sum_{i=0}^{2^k-1} P(I(x,y) = i)\log_2 P(I(x,y) = i) \quad (17)$$

where the probability of an element is denoted by $P(\cdot)$. For the gray image encryption approach, the information entropy should be as close to 8 as possible. Ten images are chosen, and the corresponding information entropy prior to and following encryption are determined using the algorithm above, as shown in Table 3. As can be observed, the picture's information entropy was lower than 7.5 before encryption, with an average of roughly 7.1. After encryption, the information entropy can reach 7.9993. This indicates that the probability of the encrypted image appearing in each pixel is very close, and the value of pixel is very random. We determine the average information entropy value and choose pertinent algorithms for comparison in terms of information entropy [17,18,43], in order to further highlight the benefits of the proposed approach. Table 4 presents the outcomes. Our algorithms all rank higher than the pertinent algorithms, demonstrating the algorithm's high degree of unpredictability.

Table 3. Different image information entropy.

Image	Entropy	
	Plaintext	Ciphertext
Man	7.1926	7.9992
Zelda	7.2668	7.9992
Einstein	6.8667	7.9992
Lena	7.4455	7.9994
Mandrill	7.3899	7.9991
Bridge	5.7056	7.9993
Crowd	7.4842	7.9994
Boat	7.1914	7.9994
Peppers	7.5936	7.9993

Table 4. Comparison with other schemes.

Algorithms	Li et al. [17]	Xu et al. [18]	Zhou et al. [43]	Ours
Cipher	7.9935	7.9972	7.9973	7.9994

4.4. Key Sensitivity Analysis

Key sensitivity analysis is the process of recovering the original image and assessing how small key changes can impact the encrypted image. Usually, changes are made to the encryption key and decryption key that are used during the encryption and decryption procedures. It is preferable to completely omit any plaintext data when using a strong key sensitivity encryption algorithm. So, we add 10−14 to x_0 and xx_0, respectively, in the encryption stage, and the encryption results are displayed in Figure 7a,b. You can see that a tiny change in the key has a significant impact on the encryption outcome. Similarly, we increase xx_1 and yy_1 by 0.01 when decrypting, and (c) and (d) of Figure 7 display the encryption results. The wrong key cannot decrypt any plaintext information. This is because the encryption technique was designed using SHA-256 and chaotic systems, both of which have a high degree of unpredictability and are particularly sensitive to beginning values. The proposed algorithm has great security as a result.

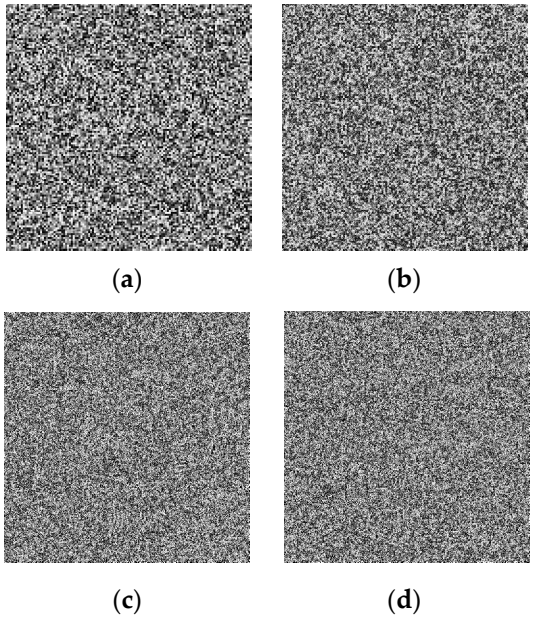

Figure 7. Key sensitivity analysis. (**a**) $x_0 + 10^{-14}$; (**b**) $xx_0 + 10^{-14}$; (**c**) $xx_1 + 10^{-14}$; and (**d**) $yy_1 + 10^{-14}$.

We also assess the algorithm's primary sensitivity from a different perspective. The ratio of two photographs with identical pixel values in the same position is known as the number of pixel change (NPCR). But, this indicator has some flaws. If the pixel values in the same position of two images are not equal but the difference is very small, human eyes have trouble distinguishing the differences. Unified average changing intensity (UACI) is used to assess how much two photographs' pixel values differ from one another. The following are NPCR and UACI [44]:

$$D(x,y) = \begin{cases} 0, C_1(x,y) \neq C_2(x,y) \\ 1, C_1(x,y) = C_2(x,y) \end{cases} \tag{18}$$

$$\text{NPCR} = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} D(x,y) \times 100 \tag{19}$$

$$\text{UACI} = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} \frac{C_1(x,y) - C_2(x,y)}{2^b - 1} \times 100 \tag{20}$$

where M,N are the image's height and breadth, respectively. A pixel from two photos is represented by $C_1(x, y)$ and $C_2(x, y)$. The effectiveness of the encryption algorithm to fend off differential attacks increases as the algorithm gets closer to its theoretical value. To compute the NPCR and UACI values throughout the encryption process, we employ the key change methodology mentioned above. To create two cipher pictures for the encryption process, the values x_0, xx_0, and yy_1 are multiplied by 10^{-14}. Table 5 displays the NPCR and UACI values before and after each parameter adjustment for the two encrypted images. The NPCR and UACI values are close to the predicted values. In conclusion, Table 5 and Figure 7 demonstrate the great key sensitivity of the picture encryption method.

Table 5. The UACIs and NPCRs between cipher pictures produced by the right ciphers and slightly different keys.

Images		Man	Peppers	Bridge	Ideal
$xx_0 + 10^{-14}$	NPCR	99.6213	99.5983	99.6342	99.6094
	UACI	33.3946	33.5314	33.5106	33.4635
$x_0 + 10^{-14}$	NPCR	99.6007	99.5918	99.6455	99.6094
	UACI	33.3146	33.6196	33.4489	33.4635
$yy_1 + 10^{-14}$	NPCR	99.6512	99.5392	99.6411	99.6094
	UACI	33.5102	33.3393	33.5161	33.4635

4.5. Differential Attack Analysis

The simulation changes the plaintext image at random, after which the two images are encrypted to produce the equivalent ciphertext image. Finally, using Equations (16) and (17), the NPCR and UACI values between the two photos were determined. The findings are shown in Table 5. Therefore, the method may successfully fend off differential assaults, selected plaintext assaults, and well-known plaintext assaults. This is due to the fact that we employ a number of strategies to enhance the correlation. The starting parameters of chaotic systems, which are extremely sensitive to initial values, are calculated using the SHA-256 hash value.

5. Conclusions

An effective encryption technique based on integer wavelet transform and cyclic shift is suggested in this research. Only four low-frequency matrices that contain a significant amount of information are jumbled when the plaintext image is modified by a two-level integer wavelet, significantly reducing the amount of information. In order to guarantee high efficiency and boost the algorithm's security, the cyclic shift is also added to the bidirectional diffusion stage. In the permutation phase and diffusion phase, many chaotic systems are used, and the user definition and SHA256 are used to establish the beginning values. We evaluate the proposed algorithm's performance across a range of metrics and contrast it with related algorithms. Our scheme offers excellent security, strong plaintext sensitivity, and the capacity to withstand statistical attacks and differential attacks, according to simulation findings.

In some scenarios, it is sometimes necessary to transfer multiple images at the same time. Therefore, it is worthwhile to propose a multi-graph encryption algorithm to ensure its security. We hope to propose multi-graph encryption algorithms with high encryption efficiency. In addition, transform domains and hyper-chaos systems are the main techniques we will focus on. We also hope to use better performance transform domains and chaotic systems in future studies. Alternatively, we can improve the existing transform or chaotic system and apply it to the encryption algorithm to further improve the algorithm performance.

Author Contributions: Conceptualization, Y.-M.L.; Methodology, D.W.; Validation, D.W.; Formal analysis, Y.-M.L.; Investigation, Y.D.; Resources, M.J.; Data curation, Y.D. and M.J.; Writing—original draft, Y.D. and M.J.; Writing—review & editing, Y.-M.L. and D.W.; Supervision, D.W.; Funding acquisition, Y.-M.L. and D.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62371364 and in part by the Natural Science Basic Research Program of Shaanxi (Program No. 2023-JC-YB-048).

Data Availability Statement: Data are contained within the article.

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

References

1. Zhou, N.; Wang, Y.; Gong, L. Novel optical image encryption scheme based on fractional mellin transform. *Opt. Commun.* **2011**, *284*, 3234–3242. [CrossRef]
2. Wei, D.; Li, Y. Convolution and multichannel sampling for the offset linear canonical transform and their applications. *IEEE Trans. Signal Process.* **2019**, *67*, 6009–6024. [CrossRef]
3. Wei, D.; Wang, R.; Li, Y.-M. Random discrete linear canonical transform. *J. Opt. Soc. Am. A-Opt. Image Sci. Vis.* **2016**, *33*, 2470–2476. [CrossRef] [PubMed]
4. Nezhad, S.Y.D.; Safdarian, N.; Zadeh, S.A.H. New method for fingerprint images encryption using DNA sequence and chaotic tent map. *Optik* **2020**, *224*, 165661. [CrossRef]
5. Wei, D.; Jiang, M.; Deng, Y. A secure image encryption algorithm based on hyper-chaotic and bit-level permutation. *Expert Syst. Appl.* **2022**, *213*, 119074. [CrossRef]
6. Wei, D.; Jiang, M. A fast image encryption algorithm based on parallel compressive sensing and DNA sequence. *Optik* **2021**, *238*, 166748. [CrossRef]
7. Yu, J.; Xie, W.; Zhong, Z.; Wang, H. Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation. *Chaos Solitons Fractals* **2022**, *162*, 112456. [CrossRef]
8. Wang, X.; Xu, D. A novel image encryption scheme based on brownian motion and pwlcm chaotic system. *Nonlinear Dyn.* **2014**, *75*, 345–353. [CrossRef]
9. Rakheja, P.; Vig, R.; Singh, P. Double image encryption using 3d lorenz chaotic system, 2D non-separable linear canonical transform and qr decomposition. *Opt. Quantum Electron.* **2020**, *52*, 1–21. [CrossRef]
10. Zhang, H.; Wang, X.-Q.; Sun, Y.-J.; Wang, X.-Y. A novel method for lossless image compression and encryption based on lwt, spiht and cellular automata. *Signal Process. Image Commun.* **2020**, *84*, 115829. [CrossRef]
11. Shao, Z.; Liu, X.; Yao, Q.; Qi, N.; Shang, Y.; Zhang, J. Multiple-image encryption based on chaotic phase mask and equal modulus decomposition in quaternion gyrator domain. *Signal Process. Image Commun.* **2020**, *80*, 115662. [CrossRef]
12. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284. [CrossRef]
13. Munoz-Guillermo, M. Image encryption using q-deformed logistic map. *Inf. Sci.* **2021**, *552*, 352–364. [CrossRef]
14. Xuejing, K.; Zihui, G. A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. *Signal Process. Image Commun.* **2020**, *80*, 115670. [CrossRef]
15. Saljoughi, A.S.; Mirvaziri, H. A new method for image encryption by 3d chaotic map. *Pattern Anal. Appl.* **2019**, *22*, 243–257. [CrossRef]
16. Arroyo, D.; Rhouma, R.; Alvarez, G.; Li, S.; Fernandez, V. On the security of a new image encryption scheme based on chaotic map lattices. *Chaos* **2008**, *18*, 33112. [CrossRef]
17. Li, Y.; Wang, C.; Chen, H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. *Opt. Lasers Eng.* **2017**, *90*, 238–246. [CrossRef]
18. Xu, Q.; Sun, K.; Cao, C.; Zhu, C. A fast image encryption algorithm based on compressive sensing and hyperchaotic map. *Opt. Lasers Eng.* **2019**, *121*, 203–214. [CrossRef]
19. Luo, Y.; Lin, J.; Liu, J.; Wei, D.; Cao, L.; Zhou, R.; Cao, Y.; Ding, X. A robust image encryption algorithm based on chua's circuit and compressive sensing. *Signal Process.* **2019**, *161*, 227–247. [CrossRef]
20. Ghazanfaripour, H.; Broumandnia, A. Designing a digital image encryption scheme using chaotic maps with prime modular. *Opt. Laser Technol.* **2020**, *131*, 106339. [CrossRef]
21. Wang, X.; Liu, C.; Jiang, D. Visually meaningful image encryption scheme based on new-designed chaotic map and random scrambling diffusion strategy. *Chaos Solitons Fractals* **2022**, *164*, 112625. [CrossRef]
22. Huo, D.; Zhu, Z.; Wei, L.; Han, C.; Zhou, X. A visually secure image encryption scheme based on 2d compressive sensing and integer wavelet transform embedding. *Opt. Commun.* **2021**, *492*, 126976. [CrossRef]
23. Rakheja, P.; Singh, P.; Vig, R. An asymmetric image encryption mechanism using qr decomposition in hybrid multi-resolution wavelet domain. *Opt. Lasers Eng.* **2020**, *134*, 106177. [CrossRef]
24. Shafique, A.; Ahmed, F. Image encryption using dynamic s-box substitution in the wavelet domain. *Wirel. Pers. Commun.* **2020**, *115*, 2243–2268. [CrossRef]

25. Li, X.; Meng, X.; Yang, X.; Wang, Y.; Yin, Y.; Peng, X.; He, W.; Dong, G.; Chen, H. Multipleimage encryption via lifting wavelet transform and xor operation based on compressive ghost imaging scheme. *Opt. Lasers Eng.* **2018**, *102*, 106–111. [CrossRef]
26. Zhang, L.; Wei, D. Image watermarking based on matrix decomposition and gyrator transform in invariant integer wavelet domain. *Signal Process.* **2020**, *169*, 107421. [CrossRef]
27. Zhang, L.; Wei, D. Robust and reliable image copyright protection scheme using downsampling and block transform in integer wavelet domain. *Digit. Signal Process.* **2020**, *106*, 102805. [CrossRef]
28. Shakir, H.R. An image encryption method based on selective aes coding of wavelet transform and chaotic pixel shuffling. *Multimed. Tools Appl.* **2019**, *78*, 26073–26087. [CrossRef]
29. An, F.-P.; Liu, J.-E. Image encryption algorithm based on adaptive wavelet chaos. *J. Sens.* **2019**, *2019*, 2768121. [CrossRef]
30. Wang, X.; Zhu, X.; Wu, X.; Zhang, Y. Image encryption algorithm based on multiple mixed hash functions and cyclic shift. *Opt. Lasers Eng.* **2018**, *107*, 370–379. [CrossRef]
31. Li, Z.; Peng, C.; Tan, W.; Li, L. A novel chaos-based color image encryption scheme using bit-level permutation. *Symmetry* **2020**, *12*, 1497. [CrossRef]
32. Qian, K.; Feng, W.; Qin, Z.; Zhang, J.; Luo, X.; Zhu, Z. A novel image encryption scheme based on memristive chaotic system and combining bidirectional bit-level cyclic shift and dynamic DNA-level diffusion. *Front. Phys.* **2022**, *718*, 963795. [CrossRef]
33. Wang, X.-Y.; Gu, S.-X.; Zhang, Y.-Q. Novel image encryption algorithm based on cycle shift and chaotic system. *Opt. Lasers Eng.* **2015**, *68*, 126–134. [CrossRef]
34. Sweldens, W. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.* **1996**, *3*, 186–200. [CrossRef]
35. Sweldens, W. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.* **1998**, *29*, 511–546. [CrossRef]
36. Li, D.; Liu, Y.; Gong, L. Color image encryption algorithm based on chua's circuit and chen's hyper-chaotic system. *J. Inf. Comput. Sci.* **2015**, *12*, 1021–1028. [CrossRef]
37. Tian, J.; Lu, Y.; Zuo, X.; Liu, Y.; Qiao, B.; Fan, M.; Ge, Q.; Fan, S. A novel image encryption algorithm using pwlcm map-based cml chaotic system and dynamic DNA encryption. *Multimed. Tools Appl.* **2021**, *80*, 32841–32861. [CrossRef]
38. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [CrossRef]
39. Zhou, N.; Zhang, A.; Zheng, F.; Gong, L. Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing. *Opt. Laser Technol.* **2014**, *62*, 152–160. [CrossRef]
40. Zhou, N.; Zhang, A.; Wu, J.; Pei, D.; Yang, Y. Novel hybrid image compression–encryption algorithm based on compressive sensing. *Opt.-Int. J. Light Electron. Opt.* **2014**, *125*, 5075–5080. [CrossRef]
41. Zhou, N.; Li, H.; Wang, D.; Pan, S.; Zhou, Z. Image compression and encryption scheme based on 2D compressive sensing and fractional mellin transform. *Opt. Commun.* **2015**, *343*, 10–21. [CrossRef]
42. Awad, A.; Awad, D. Efficient image chaotic encryption algorithm with no propagation error. *Etri J.* **2010**, *32*, 774–783. [CrossRef]
43. Zhou, K.; Fan, J.; Fan, H.; Li, M. Secure image encryption scheme using double random-phase encoding and compressed sensing. *Opt. Laser Technol.* **2020**, *121*, 105769. [CrossRef]
44. Lu, Q.; Zhu, C.; Deng, X. An efficient image encryption scheme based on the lss chaotic map and single s-box. *IEEE Access* **2020**, *8*, 25664–25678. [CrossRef]