fractal and fractional

MDPI

Article

A Novel Image Encryption Algorithm Based on Compressive Sensing and a Two-Dimensional Linear Canonical Transform

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

School of Mathematics and Statistics, Xidian University, Xi'an 710071, China; jmj20180101@163.com (M.J.); dywei@xidian.edu.cn (D.W.); dy3581908802@163.com (Y.D.)
* Correspondence: ymli@xidian.edu.cn

Abstract: In this paper, we propose a secure image encryption method using compressive sensing (CS) and a two-dimensional linear canonical transform (2D LCT). First, the SHA256 of the source image is used to generate encryption security keys. As a result, the suggested technique is able to resist selected plaintext attacks and is highly sensitive to plain images. CS simultaneously encrypts and compresses a plain image. Using a starting value correlated with the sum of the image pixels, the Mersenne Twister (MT) is used to control a measurement matrix in compressive sensing. Then, the scrambled image is permuted by Lorenz's hyper-chaotic systems and encoded by chaotic and random phase masks in the 2D LCT domain. In this case, chaotic systems increase the output complexity, and the independent parameters of the 2D LCT expand the key space of the suggested technique. Ultimately, diffusion based on addition and modulus operations yields a cipher-text image. Simulations showed that this cryptosystem was able to withstand common attacks and had adequate cryptographic features.

Keywords: optical encryption; compressive sensing; two-dimensional linear canonical transform; Lorenz's hyper-chaotic map; chaotic random phase map

Citation: Li, Y.-M.; Jiang, M.; Wei, D.; Deng, Y. A Novel Image Encryption Algorithm Based on Compressive Sensing and a Two-Dimensional Linear Canonical Transform. *Fractal Fract.* **2024**, *8*, 92. https://doi.org/10.3390/fractalfract8020092

Academic Editor: Ahmed I. Zayed

Received: 20 November 2023
Revised: 14 December 2023
Accepted: 19 January 2024
Published: 31 January 2024

1. Introduction

The rapid development of the internet has brought convenience to daily life, but privacy leakage incidents occur frequently. The issue of security has garnered extensive attention. Optical encryption, compressed sensing (CS), and network learning are effective means of protecting image security [1–3]. In 1995, Refregier and Javidi introduced the double random algorithm [4], which utilizes four optical processors and has garnered significant attention. To enhance security and increase the range of possible keys, the DRPE technique has been expanded into additional domains. However, it needs to be pointed out that these double-random phase-encoding (DRPE)-based encryption systems are linear systems [4–7], and these encryption methods are all types of symmetrical encryption systems. Due to the inherently linear nature of mathematics and optical transformation, the vulnerability of various encryption schemes to plaintext assaults is rather high [8].

Consequently, many nonlinear optical image encryption systems that can enhance security and resist plaintext attacks have been recently put forward [9–12]. An image encryption technique based on a chaotic system and transform was presented by Zhou et al. [9]. FRMTs of varying orders alter distinct annular areas of the original image, and this can be used to overcome the disadvantage of using linear transforms and having large key space. Since then, both signal processing and encryption have involved the more flexible linear canonical transform (LCT), which has three free parameters [13,14]. On the basis of the LCT, Wei et al. proposed the random discrete linear regular transform (RDLCT) [13]. The randomness of the RDLCT's output phase and magnitude can strengthen an encryption system's security. Recent designs have included further techniques based on the 2D LCT [14,15]. The 2D LCT has the natural advantage of multiple parameters, which can expand the key space and improve the security of an algorithm.

Chaos-based encryption schemes have frequently been utilized to further increase security [16–20]. This is because of their significant sensitivity to beginning circumstances, good pseudo-randomness, and ergodicity. In 1998, Fridrich proposed the classic framework of image encryption [16]. An image encryption technique based on spatiotemporal chaos was created using the framework mentioned above [17]. However, both of them were proven to be unable to resist chosen plaintext attacks [19,21]. Image encryption systems use an increasing amount of high-dimensional chaos due to their increasingly complicated dynamic nature, and they are more sensitive to initial values and higher security than to low-dimensional chaos [22]. Saljoughi presented an encryption algorithm using three-dimensional logistic maps [23]. A scale-invariant digital image encryption technique based on chaotic 3D maps was introduced by Hamed et al. [24], and it had excellent security capacity and high efficiency. Additionally, a number of hyper-chaotic systems were incorporated to enhance security [25–27]. To sum up, hyper-chaotic systems provide sequences with high pseudo-randomness that are ideal for encryption.

Meanwhile, CS is widely used because it allows data to be compressed directly at the time of collection, rather than being transmitted or stored after collection [28–30]. This can reduce the amount of data transferred and stored, increasing efficiency. At the same time, the nature of random measurements makes it more difficult to reconstruct information from raw data, which can increase the strength of encryption. Measurement matrices are typically thought of as the key in compressive-sensing-based algorithms [31–33]. However, the most popular approach generates an image-encrypted measurement matrix using a chaotic system [31,32,34–36]. Zhou developed an approach that employed two measurement matrices under the control of a logistic map to compress and encrypt pictures [31]. Nevertheless, the aforementioned techniques' secret keys were unrelated to plaintext pictures, making them susceptible to specific plaintext assaults. Some recent methods can reduce these problems. The work in [37] introduced parallel compressive sensing (PCS), which was resistant to specific plaintext assaults. Above all, we created a unique image encryption technique based on CS and 2D LCT that could withstand typical assaults and had a huge key space.

In this study, we introduced a unique image encryption technique based on CS and the 2D LCT to further increase security. First, CS offers trustworthy and effective encryption and compression services, and a plaintext image controls the CS measurement matrix's construction parameters. Second, to prevent certain plaintext assaults, the hyper-chaotic system's initial values were generated by using the SHA 256 value of the original image. Third, the 2D LCT had six free parameters, which greatly enhanced the key space. The suggested scheme's security against various assaults was demonstrated by the findings in security analysis and testing.

The following is the format for the remainder of this paper: Section 2 presents a few foundational theories. Section 3 provides a detailed explanation of the recommended image encryption method. A number of presentations and simulations are presented in Section 4. In Section 5, a succinct summary of the findings is provided.

2. Preliminaries

We first give a few related definitions before going into detail about the suggested algorithm.

2.1. Linear Canonical Transform

The linear canonical transform (LCT) is a class of linear integral transformations with three parameters, which is the general case of FrFT and Fresnel transformations (FSTs) [38]. The LCT is also useful in optical signal processing and encryption in a number of ways due to its multi-parameter advantages. The LCT of a transformable signal can be defined as [39]

$$L_f^A(u) = \mathcal{L}_A[f(t)](u) = \begin{cases} \int_{\mathbb{R}} f(t) K_A(u,t) \mathrm{d}t, & b \neq 0, \\ \sqrt{d} \exp\left(\mathrm{i}\frac{cdu^2}{2}\right) f(\mathrm{d}u), & b = 0, \end{cases} \tag{1}$$

where $K_A(u,t) = C_A \exp\left(\mathrm{i}\frac{at^2}{2b} - \mathrm{i}\frac{ut}{b} + \mathrm{i}\frac{du^2}{2b}\right)$, $C_A = \frac{1}{\sqrt{\mathrm{i}2\pi b}}$, and constants a, b, c, d are related by a parameter matrix A, which is also called unit-modular matrix and is represented as

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{2}$$

where $ad - bc = 1$. Consequently, the LCT has three independent parameters.

A function's continuous 2D LCT can be expressed as

$$\begin{aligned} & F^{A,B}\{f(m,n)\} = F^{C,D}(x,y) \\ & = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(m,n)K_{C,D}(m,n,x,y)\mathrm{d}m\ \mathrm{d}n \\ & = \begin{cases} \frac{1}{j2\pi}\sqrt{\frac{1}{c_1c_2}}\mathrm{e}^{\left(\frac{d_1x^2}{2c_1}+\frac{d_2m^2}{2c_2}\right)} \\ \times \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mathrm{e}^{-j\left(\frac{xm}{c_1}+\frac{yn}{c_2}\right)+\int\left(\frac{a_1x^2}{2c_1}+\frac{a_2y^2}{2c_2}\right)} f(m,n)\mathrm{d}m\ \mathrm{d}n \\ c_1c_2 \neq 0, |C| = |D| = 1 \\ \sqrt{d_1d_2}\mathrm{e}^{\frac{\int\left(a_1d_1u^2+a_2d_2^2\right)}{2}} \quad f(d_1m, d_2n) \quad c_1^2 + c_2^2 = 0 \end{cases} \end{aligned} \tag{3}$$

where $K_C(m,x) = \sqrt{\frac{1}{j2\pi c_1}}\ \mathrm{e}^{j\frac{d_1x^2}{2c_1} - j\frac{xm}{b_1} + j\frac{a_1m^2}{2c_1}}$, $K_D(n,y) = \sqrt{\frac{1}{j2\pi c_2}}\mathrm{e}^{j\frac{d_2y^2}{2c_2} - j\frac{yn}{c_2} + j\frac{a_2x^2}{2c_2}}$, $C = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}$, and $D = \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix}$ are real matrices. $C^{-1} = \begin{bmatrix} d_1 & -b_1 \\ -c_1 & a_1 \end{bmatrix}$, $D^{-1} = \begin{bmatrix} d_2 & -b_2 \\ -c_2 & a_2 \end{bmatrix}$. The constraint condition of the matrix C and D is $|C| = |D| = 1$.

2.2. Compressive Sensing

A unique method that can accomplish compression during the sampling process is called compressive sensing (CS) [28]. One might refer to a plain picture, secret key, or cipher image for the original image, the measurement matrix, and the image measured in CS, in that order. In accordance with the CS theory, sparse signals can yield a small quantity of observation data, and by resolving an optimization issue, we can roughly reconstruct these signals.

Suppose that $x \in R^N$ is a signal and can be measured by

$$\alpha = \Psi^T x \tag{4}$$

where Φ is the sparse coefficient vector and Ψ is an orthonormal sparse basis. The measured values can be obtained through the projection of a coefficient series on Φ. After that, the following signal CS procedure can be stated:

$$Y = \Phi x = \Phi\Psi\alpha \tag{5}$$

where $y \in R^M$, $\Phi \in R^{M\times N}$, $\Theta = \Phi\Psi$. Similarly, 2D or high dimension (HD) signals can be reduced to the 1D format by superimposing column vectors.

Reconstructing the original signal x from the measurements requires figuring out how to solve the following l_0 issue:

$$\min \|\alpha\|_0 \ s.t.\ Y = \Theta\alpha \tag{6}$$

where the l_0-norm of a vector is indicated by $\|\cdot\|_0$. Several popular reconstruction techniques have been introduced [40–42].

In this work, we implement the discrete wavelet transform (DWT) to achieve sparsity in the signals. A technique for signal processing called DWT breaks a signal down into many signals with varying scales, temporal resolutions, and frequency resolutions. The next

part will provide a detailed description of how the measurement matrix was constructed. Eventually, x is restored by utilizing the orthogonal matching pursuit (OMP) method.

2.3. Lorenz's Hyper-Chaotic System

Mathematically, Lorenz's hyper-chaotic system is defined as

$$\begin{cases} \dot{x} = d(y - x) + w \\ \dot{y} = fx - y - xz \\ \dot{z} = xy - ez \\ w = -yz + rw \end{cases} \tag{7}$$

where the hyper-chaotic system's control parameters are d, e, f, and r, and $d = 10$, $e = 8/3$, $f = 28$, $r \in (-1.52, -0.06)$. Figure 1 displays the system's attractors. The average exponential divergence of neighboring trajectories in phase space is numerically represented by the Lyapunov exponent. This is one of the characteristics that helps distinguish chaotic motion. A chaotic system is often characterized by a positive maximum Lyapunov exponent. As a result, this hyper-chaotic system predicts events more quickly than a typical chaotic system does. Simultaneously, this feature makes the encryption system more secure.

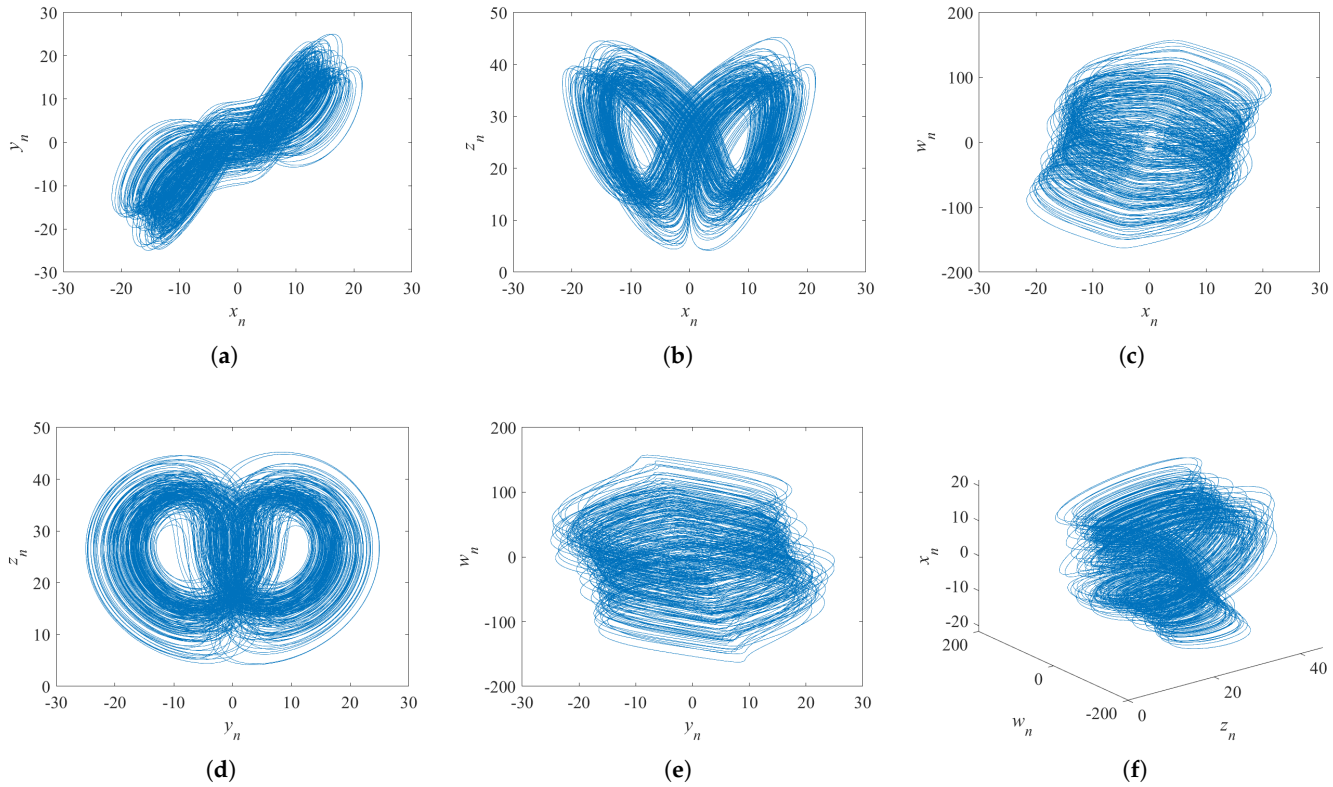


Figure 1. Lorenz's hyper-chaotic attractor: (**a**) x-y plane; (**b**) x-z plane; (**c**) x-w plane; (**d**) y-z plane; (**e**) y-w plane; (**f**) x-w-z plane.

3. The Proposed Image Encryption and Decryption Algorithm

3.1. Encryption Process

Four steps make up the majority of the suggested encryption method. The image must first be compressed and encrypted. Next, a prior image is jumbled using Lorenz's hyper-chaotic system. The encrypted image is then encrypted using a 2D LCT. The image that was processed above is diffused the final stage. The suggested algorithm's encryption procedure is depicted in Figure 2. Here, we take an image of size $M \times N$ as an example to discuss the four parts in detail.

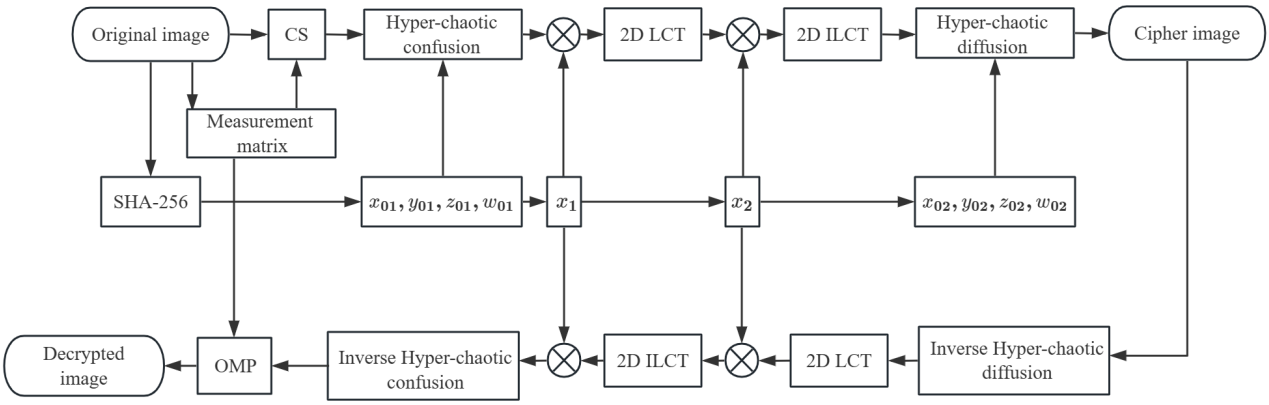


Figure 2. A flowchart of the procedure for encryption and decoding.

3.1.1. Key Generation

The cryptosystem's secret key structure has two parts. The first part is given by the encryption system user. Another part is obtained by SHA-256. Because SHA-256 is so dependent on the initial values of the original picture, even a small change might provide completely different secret keys. Therefore, we might use it to generate the keys for the suggested encryption technique, which can withstand attacks with carefully selected and well-known plaintext.

SHA-256 generates a 256-bit secret key as the hash value. This may be represented as and split up into 8-bit chunks: k_1, k_2, k_3..., k_{32} ($k_i = \{k_0^i, k_1^i, k_2^i, \ldots, k_7^i\}$). It is simultaneously converted into 32 decimal numbers k_1, k_2, k_3..., k_{32} for easier application. The following can be used to obtain the starting values.

Step 1: The parameters of Lorenz's hyper-chaotic system are follows:

$$\begin{cases} x_{01} = \mathrm{mod}\left(\mathrm{sum}(k_1, k_2, k_3, k_4)/256, 1\right) + x'_{01} \\ y_{01} = \mathrm{mod}\left(\mathrm{sum}(k_5, k_6, k_7, k_8)/256, 2\right) + y'_{01} \\ z_{01} = \mathrm{mod}\left(\mathrm{sum}(k_9, k_{10}, k_{11}, k_{12})/256, 3\right) + z'_{01} \\ w_{01} = \mathrm{mod}\left(\mathrm{sum}(k_{13}, k_{14}, k_{15}, k_{16})/256, 4\right) + w'_{01} \end{cases} \tag{8}$$

$$\begin{cases} x_{02} = \mathrm{mod}\left(\mathrm{sum}(k_{17}, k_{18}, k_{19}, k_{20})/256, 1\right) + x'_{02} \\ y_{02} = \mathrm{mod}\left(\mathrm{sum}(k_{21}, k_{22}, k_{23}, k_{24})/256, 2\right) + y'_{02} \\ z_{02} = \mathrm{mod}\left(\mathrm{sum}(k_{25}, k_{26}, k_{27}, k_{28})/256, 3\right) + z'_{02} \\ w_{02} = \mathrm{mod}\left(\mathrm{sum}(k_{29}, k_{30}, k_{31}, k_{32})/256, 4\right) + w'_{02} \end{cases} \tag{9}$$

where *sum* denotes the sum of k_1, k_2, k_3, and k_4; x_{01}, y_{01}, z_{01}, w_{01} are the starting points of permutation, and $x_{02}, y_{02}, z_{02}, w_{02}$ are the starting points of diffusion.

Step 2: The following are the starting values that are applied to chaotic random phase masks (CRPMs):

$$x_1 = \mathrm{mod}\left(\left(\begin{array}{c} \frac{k_1 \oplus \cdots \oplus k_4 + k_5 \oplus \cdots \oplus k_8}{256} \\ + \frac{k_9 \oplus \cdots \oplus k_{12} + k_{13} \oplus \cdots \oplus k_{16}}{256} \end{array}\right), 1\right) + x'_1 \tag{10}$$

$$x_2 = \mathrm{mod}\left(\left(\begin{array}{c} \frac{k_{17} \oplus \cdots \oplus k_{20} + k_{31} \oplus \cdots \oplus k_{24}}{256} \\ + \frac{k_{25} \oplus \cdots \oplus k_{28} + k_{29} \oplus \cdots \oplus k_{32}}{256} \end{array}\right), 1\right) + x'_2 \tag{11}$$

where $\oplus$ is the bitxor operation. x_1 and x_2 are the starting values of the two CRPMs.

The starting numbers are $x_{01}, y_{01}, z_{01}, w_{01}$, $x_{02}, y_{02}, z_{02}, w_{02}$, x_1, x_2, which fluctuate in tandem with the original image. $x'_{01}, y'_{01}, z'_{01}, w'_{01}, x'_{02}, y'_{02}, z'_{02}, w'_{02}$, and x'_1, x'_2 are supplied by the encryption system's user to increase security even more.

3.1.2. The Compression–Encryption Steps

First, the original picture was made sparser by applying a discrete wavelet transform. Then, to balance the sparsity of each row and improve the encryption performance, the resultant image was permuted using the Arnold transform, and this was repeated 25 times with starting values of $a = 15$ and $b = 8$. Lastly, the encrypted image was obtained by using CS, which had the ability to both compress and encrypt images.

In CS, the measurement matrix was managed by the Mersenne Twister (MT) algorithm, which is based on a twisted generalized feedback shift register and can produce high-quality pseudo-random numbers. The random sequence was reshaped to generate a measurement matrix. This was efficient and made up for the shortcomings of traditional pseudo-random number generators. The initial conditions of MT were as follows:

$$key = round\left(\frac{\sum_{x=1}^{M}\sum_{y=1}^{N} I(x,y)}{M \times N}\right) + key_0 \tag{12}$$

where the closest integer to x is indicated by $round(x)$, and $\sum_{x=1}^{M}\sum_{y=1}^{N} I(x,y)$ represents the sum of the pixel values of image I.

3.1.3. Permutation and Diffusion

To confuse and disperse pictures, Lorenz's hyper-chaotic system generated pseudo-random sequences using the starting values generated in Equation (8). The process of permutation is given in detail:

Step 1: To create a one-dimensional vector in rows or columns, the original two-dimensional image matrix P (in this paper, expanded by row), denoted as A, is expanded.

Step 2: A pseudo-random sequence X_i ($i = 1, 2, \cdots, MN$) of length $M \times N$ is generated with the aid of the Lorenz system.

Step 3: Only the first pseudo-random number that appears repeatedly in X is retained, and, in order of smallest to greatest, the values in the set $\{1, 2, \cdots, MN\}$ that are absent from X are appended to the end of X.

Step 4: The positions of $A(X_i)$ and $A(X_{MN-i+1})$ are swapped.

The diffusion algorithm in our proposed encryption based on addition and modulus operations is represented by the following equation.

$$D_i = (D_{i-1} + S_i + P_i) \bmod 256 \tag{13}$$

where S_i is the pseudo-random sequence produced by Lorenz's system, D_i represents the cypher vectors, and P_i is a vector that expands from the plaintext image. Equation (13) is expanded:

$$D_n = (D_0 + S_1 + \cdots + S_n + P_1 + \cdots + P_n) \bmod 256 \tag{14}$$

It can be found in the Equation (14) that the information of the plaintext pixel P_i can only be hidden in $D_i \sim D_N$. In order to achieve the dispersion of each plaintext pixel over the whole cipher-text picture, this approach must be run twice. The following is the diffusion process.

Step 1: Using rows or columns, a one-dimensional vector is created by expanding the two-dimensional original image matrix P, denoted as A.

Step 2: With the initial values $x_{02}, y_{02}, z_{02}, w_{02}$ of Lorenz's hyper-chaotic system from Equation (9) in the first part, we generate two pseudo-random sequences S_1 and S_2, which are used for forward diffusion and reverse diffusion, respectively.

Step 3: S_1 is used to perform forward diffusion for the image to be encrypted using Equation (13).

Step 4: S_2 is used to perform reverse diffusion for the image to be encrypted. Reverse diffusion is expressed as follows: $D_i = (D_{i+1} + S_i + P_i) \bmod 256$.

3.1.4. CRPMs and 2D LCT Encryption

The four phases involved in the encryption process utilizing the 2D LCT and CRPMs are as follows:

Step 1: $R_1(m,n)$ is the first chaotic random phase mask generated with Equation (10) using the logistic map with the starting value x_1.

Step 2: To obtain the image $I^{'}(m,n)$, the previously encrypted complex image $I(m,n)$ is modulated with the first random phase mask $R_1(m,n)$ and encrypted using the 2D LCT.

Step 3: The logistic map with an initial value of m_2 produced with Equation (11) is used to generate the second chaotic random phase mask $R_2(m,n)$.

Step 4: After modulating $R_2(m,n)$, the inverse 2D LCT (I–LCT) operation on $I^{''}(m,n)$ yields a white-noise-like image $I^{'}(m,n)$.

The encryption process using the 2D LCT and CRPMs can be simplified as follows:

$$I''(m,n) = \text{I–LCT}\{\{\text{LCT}[I(m,n) \cdot \exp[\text{i}2\pi R_1(m,n)]]\} \cdot \exp[\text{i}2\pi R_2(m,n)\} \tag{15}$$

Figure 2 provides an illustration of the decryption procedure. It is evident that the decryption process is the opposite of the encryption technique described above.

4. Simulation Results and Security Analysis

4.1. Results of Encryption and Decryption

To test the suggested encryption scheme, four grayscale pictures of 512 × 512 pixels each—"Boat," "Zelda," "Gold hill," and "Einstein"—were chosen. The following are the parameters that we utilized in the simulation: $key_0 = 10$, $a = 15$, $b = 8$, $x_{01}^{'} = 0.1$, $y_{01}^{'} = 1.2$, $z_{01}^{'} = 2.3$, $w_{01}^{'} = 3.4$, $x_{02}^{'} = 1.1$, $y_{02}^{'} = 2.2$, $z_{02}^{'} = 3.3$, $w_{02}^{'} = 4.4$, $a_1 = 0.5$, $b_1 = 1$, $c_1 = 1$, $a_2 = 0.5$, $b_2 = 2$, $c_2 = 1.5$, $x_1^{'} = 0.23$, and $x_2^{'} = 0.72$. Figure 3 presents the validation results for the encryption and decryption algorithms at CR = 0.5. The encrypted images are, as we can see, smaller than those with plaintext. This denotes a compression of the original image. In this case, the encrypted images were half the size of the original images, which allowed the encryption effect to be accomplished, in addition to lowering the quantity of data transmitted throughout the process. Furthermore, the encrypted images had a noise-like quality, indicating that the encryption technique effectively concealed the original images' content. The decoded images were about the same as the plaintext images. In order to conduct a quantitative assessment of the quality of the reconstructed picture, the peak signal-to-noise ratio (PSNR) was introduced:

$$\text{PSNR} = 10\log_{10}\frac{255^2}{\frac{1}{MN}\sum_{j=1}^{M}\sum_{k=1}^{N}[I^{'}(j,k) - I(j,k)]^2} \tag{16}$$

where the pixel values of the restored picture $I^{'}$ and the original image I are denoted by $I^{'}(j,k)$ and $I(j,k)$, respectively. The four reconstructed pictures had PSNRs of 32.1021 dB, 34.0968 dB, 31.8867 dB, and 33.5572 dB. This demonstrated the algorithm's strong decryption performance and great accuracy in image reconstruction.

The compression ratio (CR) is defined by

$$\text{CR} = \frac{I^{'}\_h \times I^{'}\_w}{I\_h \times I\_w} \tag{17}$$

where $I^{'}\_h$,I_h and $I^{'}\_w$, I_w indicate the height and width of the image, respectively. The encrypted and decrypted results for Zelda with various CRs are listed in Figure 4. Figure 4 illustrates that, despite the modest CR, the reconstructed images exhibited remarkable similarity to the original images. When CR = 0.2, the cypher image's PSNR was 30.9034 dB. Table 1 displays the PSNRs for Lena for comparison with those of other techniques [26,32,43]. It is evident that the suggested method's reconstruction outperformed that of other approaches when the CR was higher than 0.5. Moreover, at CR = 0.4, the decrypted image's PSNR might have been higher than 30 dB. Optimizing sparsity prior to compression could enhance the effect of reconstruction. In summary, our method was able to produce better results with a smaller sample size of data.

We tested the complexity of encryption and decryption algorithms when CR = 0.5, and the results are shown in Table 2. As can be seen from the results, the encryption time is very short, and the decryption time takes a few seconds. This is because the measurement matrix is easily generated during the encryption process, and during the decryption process, we used the OMP algorithm to iteratively solve it.

The mean square error (MSE) below the division sign of Equation (16) can be used to reflect the difference between the two images before and after encryption. The greater the MSE, the greater the difference between the two images. We calculated the MSE of the four images before and after encryption and summarized them in Table 3. In combination with Figure 3, the difference between the images before and after encryption is huge, indicating that our algorithm can hide image information well.

Table 1. PSNRs (256 × 256) for Lena with various CRs.

CR	Ref. [43]	Ref. [32]	Ref. [26]	Ours
0.25	22.62	26.06	26.52	24.76
0.5	26.87	29.82	29.23	33.10
0.75	30.82	29.56	29.22	33.44

Table 2. The encryption and decryption times for different images at CR = 0.5 (unit: seconds).

Image	Encryption Time	Decryption Time
Boat	0.2301	8.7842
Zelda	0.2057	9.8499
Gold hill	0.2257	7.8853
Einstein	0.2156	8.7155

4.2. Security Evaluation

We assessed the encryption system's effectiveness from many angles based on the image properties in order to confirm the security of the suggested encryption system.

4.2.1. Analysis of Key Spaces

The key space is a crucial metric for assessing a cryptosystem's quality. Furthermore, it needs to be large enough to withstand brute-force attacks [44]. The suggested algorithm's secret key, which was mostly generated via SHA-265 of the plaintext image, is explained in Section 3.1.1. The encryption method involved 19 parameters and initial keys, which were key_0 for the measurement matrix and a, b for the Arnold map in CS, $x^{'}_{01}, y^{'}_{01}, z^{'}_{01}, w^{'}_{01}$ and $x^{'}_{02}, y^{'}_{02}, z^{'}_{02}, w^{'}_{02}$ for Lorenz's hyper-chaotic system in confusion and diffusion, $a_1, b_1, c_1, a_2, b_2, c_2$ for the 2D LCT, and $x^{'}_1, x^{'}_2$ for CRPM. We used the 2D LCT and Arnold map parameters as the public key. Presuming that the precision of the computer was 10^{-14}, the key space was large compared to 2^{100}, as it was about $(10^{14})^{11} = 10^{154} \approx 2^{511}$. As a result, the suggested method was immune to all types of brute-force assaults. Furthermore, a comparison is made between

the key spaces of four encryption techniques [26,32,45]. Table 4 illustrates that our scheme had the greatest key space.

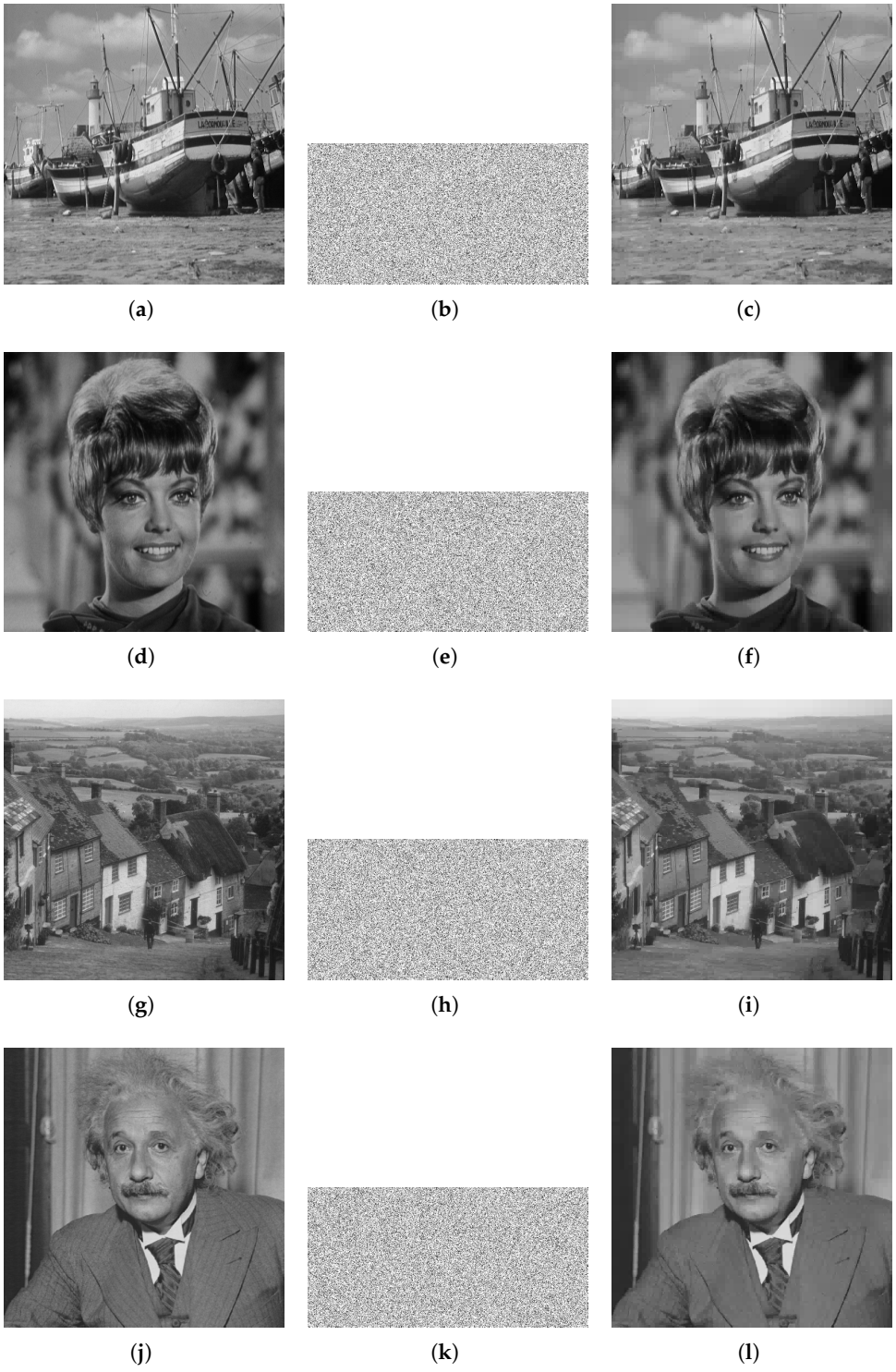


Figure 3. The results of encryption and decryption: (**a**,**d**,**g**,**j**) Unencrypted images; (**b**,**e**,**h**,**k**) the corresponding encrypted images at CR = 0.5; (**c**,**f**,**i**,**l**) the decrypted images.

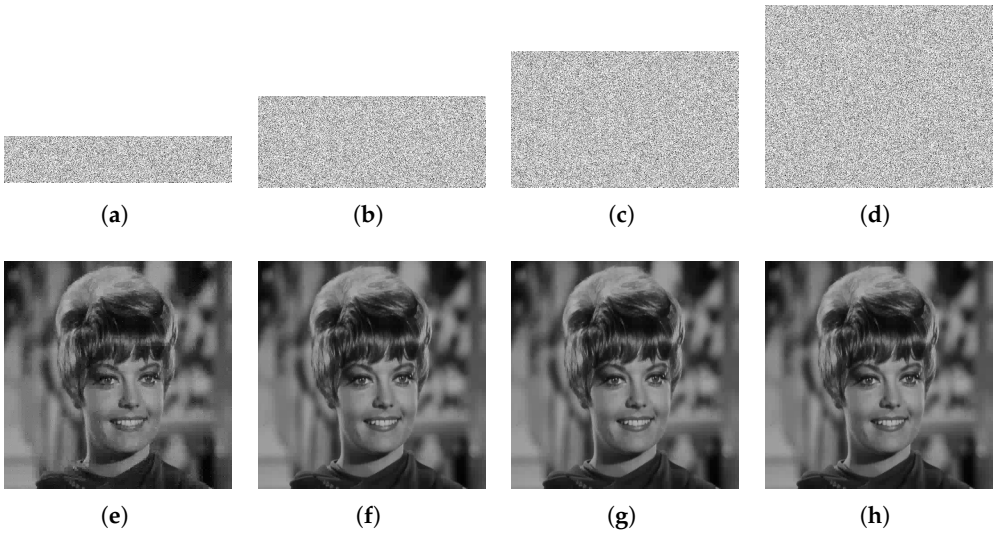


Figure 4. Results of encryption and decryption for Zelda with various CRs: (**a**–**d**) the cypher pictures for CR values of 0.2, 0.4, 0.6, and 0.8, respectively; (**e**–**h**) the corresponding decrypted images.

Table 3. MSE between plaintext images and cipher-text images at CR = 0.5.

Image	MSE
Boat	48,196.2
Zelda	43,602.8
Gold hill	47,532.0
Einstein	45,190.7

Table 4. Comparison of several approaches' key spaces.

Algorithms	Ref. [26]	Ref. [32]	Ref. [45]	Ours
Key Space	2^{299}	2^{232}	10^{102}	2^{511}

4.2.2. Histogram Analysis

A histogram provides statistical information in addition to displaying the distribution of pixels in an image. The even and smooth distribution of a cipher-text histogram can prevent attackers from breaking the image with statistical pixel value analysis. The histograms of the four images—"Boat", "Zelda", "Gold hill", and "Einstein"—at CR = 0.75, each with 512 × 512 pixels, are provided in Figure 5. As can be seen in Figure 5a–d, the histograms were often unevenly distributed and carried certain features of plaintext images. Figure 5e–h shows that the probability of pixels in the cipher-text image appearing in any value in the range of 0–255 was basically the same, and the height was approximately a straight line. After decrypting the cipher-text image with a decryption algorithm, the calculated histogram was similar to that of the plaintext image, indicating that our image decryption method recoverde the image information well.

More precisely, the distance of the histogram represents the similarity between a plaintext image and an encrypted image. For grayscale images, this is defined as the sum of the minimum heights of any pixel of the two histograms from 0 to 255, divided by the sum of the heights of all pixels of one of the histograms. The distance of the histogram is between 0 and 1, and the closer to 1, the more similar the two are. We calculated the histogram distance in the four selected graphs, and the results are shown in Table 5 and compared with those in the relevant literature. Although our algorithm was slightly worse than that in Ref. [26], the distance value was able to reach more than 0.9, indicating that

our algorithm could also recover the original image well. In addition, according to Figure 5, our algorithm was able to resist statistical attacks well.

Table 5. The histogram distance between a plaintext image and a decrypted image.

Image	Ref. [30]	Ours
Boat	0.9520	0.9522
Zelda	-	0.9586
Gold hill	0.9634	0.9625
Einstein	0.9578	0.9552

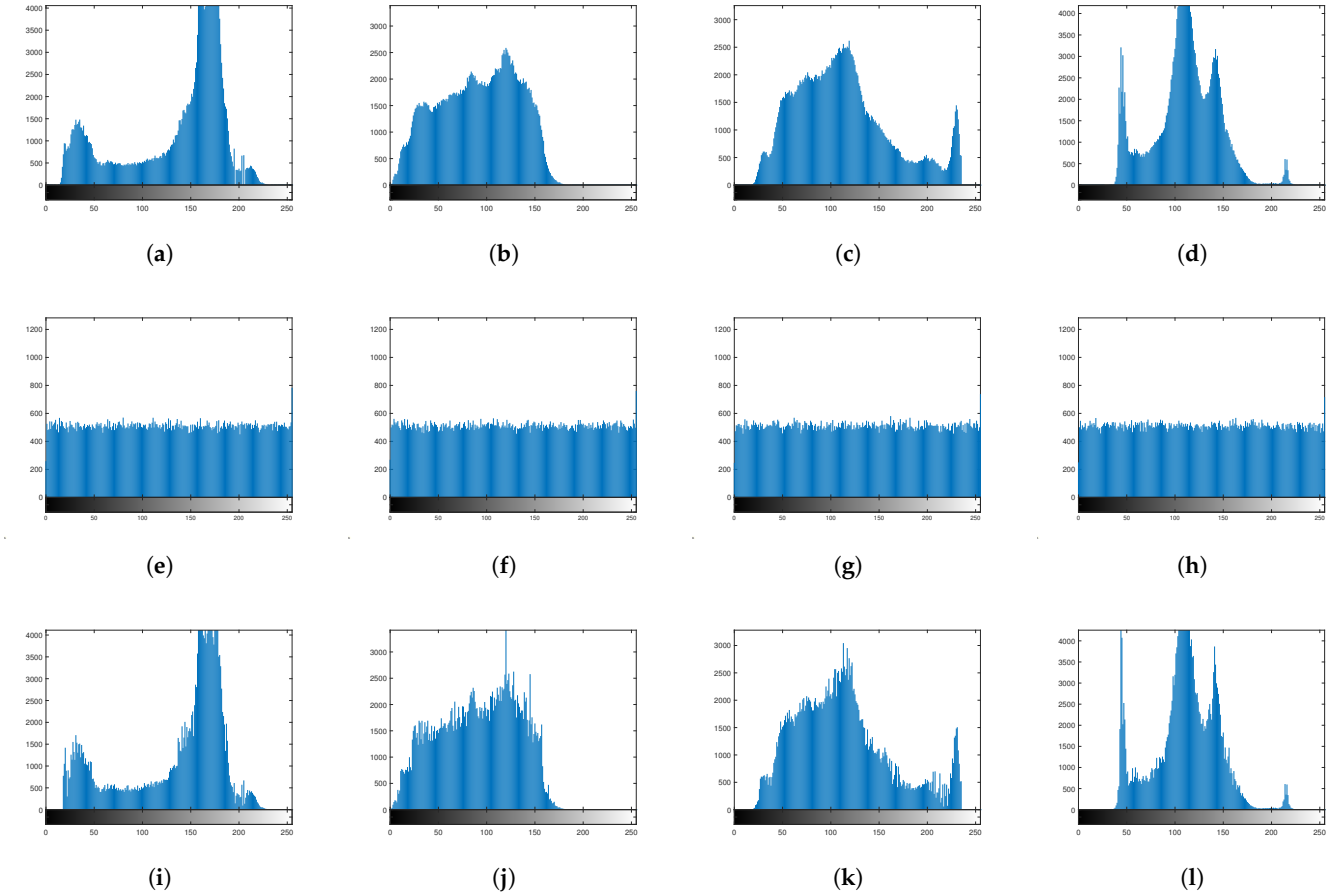


Figure 5. Histogram analysis: The histograms of the plaintext images are shown in (**a**–**d**); the ground histograms of the cipher-text images are shown in (**e**–**h**); the histograms of the decrypted images are shown in (**i**–**l**).

4.2.3. Analysis of Correlation Coefficients

Plaintext images can be attacked because of the significant correlations between their neighboring pixels. A strong method for image encryption can completely destroy the correlations. The measurement of correlation coefficients between plaintext and cipher-text images is a commonly used method for the qualitative assessment of encryption algorithms. This can be defined as [34]

$$r_{ij} = \frac{\mathrm{cov}(i,j)}{\sqrt{D(i)D(j)}} \tag{18}$$

$$\mathrm{cov}(i,j) = E[i - E(i)][j - E(j)] \tag{19}$$

where the standard variances of the values are represented by $D(\cdot)$, and the average values of the image pixels are indicated by $E(\cdot)$. There are 0 to 1 correlation coefficients. The link is greater the closer it gets to 1.

In this study, we computed the correlations from three directions at random locations. Table 6 and Figure 6 present the correlation coefficient performance on several images that were encrypted using the suggested approach. The plaintext images' pixel coordinates were clearly clustered close to the diagonal, indicating a strong correlation coefficient. Nevertheless, the cipher-text images' pixel locations filled the whole coordinate space, and the correlation coefficient dropped to almost zero. Additionally, using various encryption techniques, Table 7 contrasts Lena's correlation coefficients with 256 × 256 pixels [31,34,43]. In the cipher-text images, there was essentially no connection between neighboring pixels, suggesting that the suggested approach may entirely eliminate pixel correlation and fend off statistical assaults.

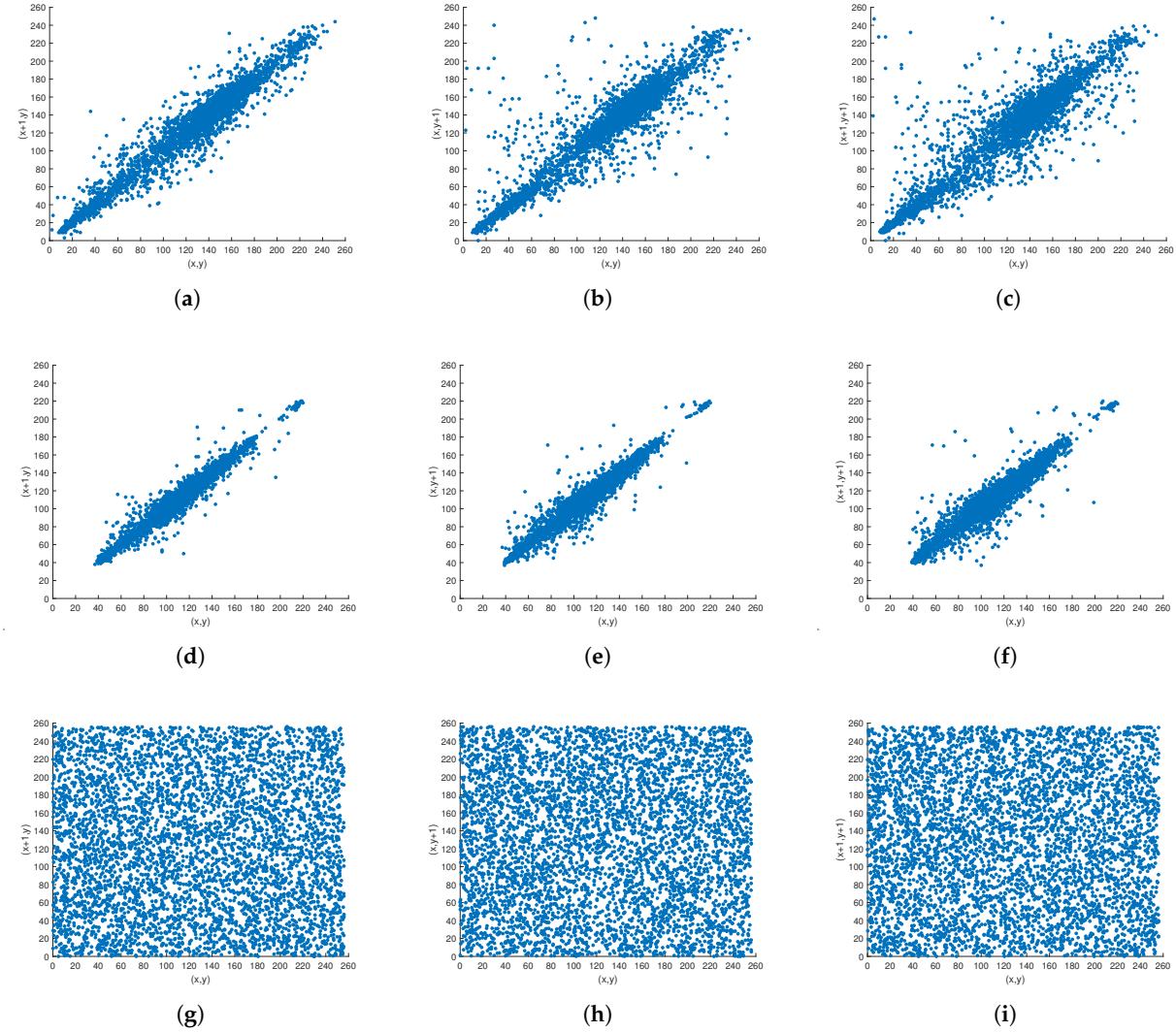
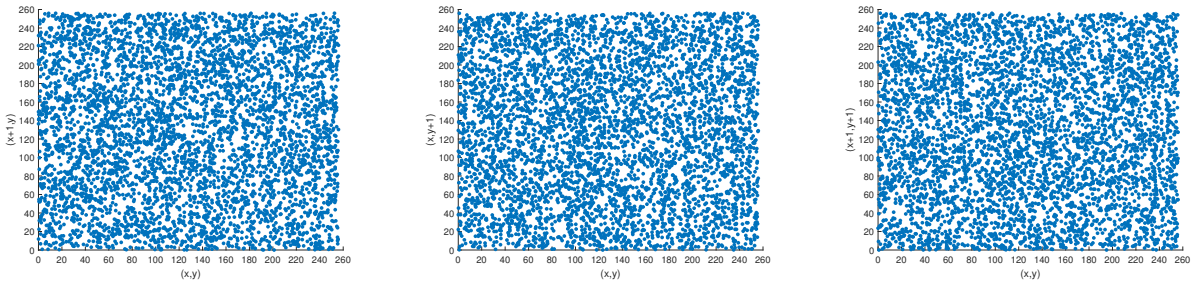


Figure 6. *Cont.*



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

Figure 6. Correlation coefficient analysis: (**a**–**c**) Einstein's correlations from three directions; (**g**–**i**) the corresponding encrypted image correlations; (**d**–**f**) Boat's correlations from three directions; (**j**–**l**) the corresponding encrypted image correlations.

Table 6. Correlation between neighboring pixels in various images.

Images	Horizontal	Vertical	Diagonal
Boat	0.9383 0.0175	0.9729 −0.0022	0.9250 −0.0089
Zelda	0.9829 −0.0061	0.9917 0.01533	0.9783 0.0035
Einstein	0.9750 0.0142	0.9801 0.0182	0.9588 −0.005
Peppers	0.9803 −0.0101	0.9811 0.0011	0.9699 −0.0094
Man	0.9631 0.0094	0.9700 0.0103	0.9453 −0.0037
Gold hill	0.9718 0.0056	0.9729 −0.0093	0.9525 −0.0282
Couple	0.9480 −0.0027	0.9511 0.0081	0.9124 0.0097

Table 7. Correlation between neighboring Lena pixels using several techniques.

Direction	Horizontal	Vertical	Diagonal
Lena (256 × 256)	0.9706	0.9853	0.9588
Ref. [34]	0.0846	0.0583	0.0931
Ref. [43]	0.0198	0.0141	0.0026
Ref. [31]	0.0104	0.0299	0.0062
Ours	0.0071	0.0121	−0.0073

4.2.4. Analysis of Information Entropy

An image I's information entropy is described as [46]

$$H(I) = -\sum_{i=0}^{2^k-1} P(I_{(x,y)=i}) \log_2 P(I_{(x,y)=i}) \tag{20}$$

where $P(\cdot)$ represents an element's chance of occurring. The information entropies of several images are shown in Table 8. The entropy in the encrypted Lena image was compared with those in Refs. [25,26,47], as displayed in Table 9. It appears that the entropy values for several hidden images were around 8. From this, we may infer that the encrypted

image data had a high degree of unpredictability and that the suggested approach could fend off statistical assaults.

Table 8. Entropy of information for several images.

Image	Entropy	
	Plaintext	**Ciphertext**
Boat	7.1914	7.9971
Zelda	7.2668	7.9973
Einstein	6.8667	7.9974
Peppers	7.5936	7.9968
Man	7.1926	7.9972
Gold hill	7.4778	7.9971
Couple	7.0581	7.9973

Table 9. Comparison the information entropy.

Methods	**Ref. [26]**	**Ref. [25]**	**Ref. [47]**	**Ours**
Cipher	7.9935	7.9972	7.9973	7.9974

4.2.5. Analysis of Cropping Attacks

During the image transfer procedure, data loss frequently occurs. A restored image can be significantly impacted by the loss of encrypted image data. Cropping attacks should be thwarted by a strong cryptosystem. Figure 7 displays a cypher image cropped at various sizes and places, whereas Figure 7 illustrates the matching reconstructed images. We conclude that while cropping affects the restoration of an image, the original image may still be recognized. Additionally, the quality of the recovered image deteriorates as the cropping area increases. As a result, our technique is resistant to a variety of cropping assaults.

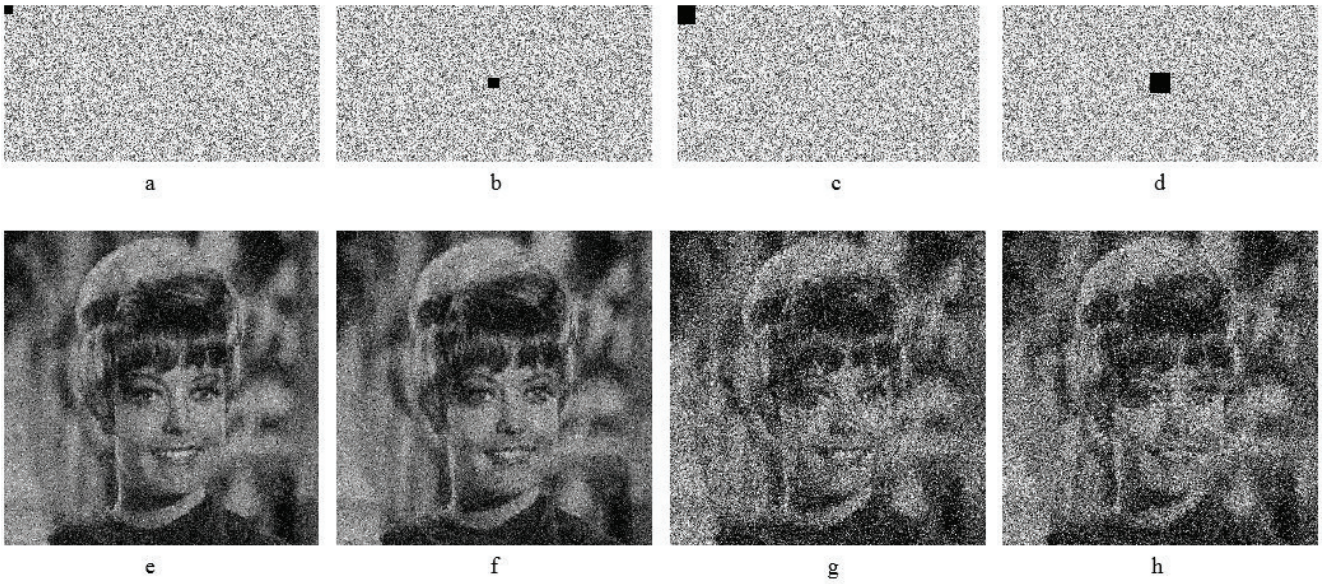


Figure 7. Cropping attack analysis: (**a**–**d**) are encrypted images with data loss in different size and position; (**e**–**h**) are corresponding decrypted images.

4.2.6. Crucial Sensitivity Testing

A safe cryptosystem is extremely sensitive to private keys. Changes in the cypher image and recovered image can be significant even with a small modification in the secret key. To subjectively assess the decryption process's key sensitivity, we changed each of x'_{01},y'_{01} for permutation, x'_{02},y'_{02} for diffusion, x'_0 for CRPM, and a_1 for the 2D LCT by adding 10^{-14}. Figure 8 displays the findings on key sensitivity during the decryption procedure. As Figure 8 illustrates, minor changes in the keys would not allow the original image to be accurately reconstructed, and we did not get any original information from the plaintext images.

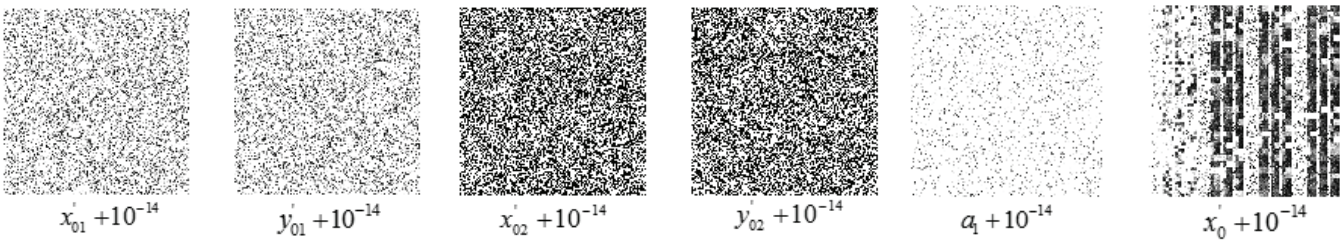


Figure 8. Key sensitivity analysis. A decrypted image was obtained by slightly changing the keys: x'_{01}, y'_{01} for permutation, x'_{02}, y'_{02} for diffusion, and x'_0, a_1 for CRPM and the 2D LCT.

Furthermore, the difference between two images is frequently measured by using the unified average change intensity (UACI) and number of pixels changed (NPCR). The following are the definitions of the NPCR and UACI between two distinct images [48]:

$$D(m,n) = \begin{cases} 0, L_1(m,n) = L_2(m,n) \\ 1, L_1(m,n) \neq L_2(m,n) \end{cases} \quad (21)$$

$$\text{NPCR} = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{n=1}^{N} P(m,n) \times 100 \quad (22)$$

$$\text{UACI} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{L_1(m,n) - L_2(m,n)}{2^b - 1} \times 100 \quad (23)$$

One pixel each of the two images is represented by $L_1(m,n)$ and $L_2(m,n)$. We used the key change method proposed above to calculate NPCR and UACI in the encryption process. For the encryption process, two cipher images were obtained by changing x'_{01}, y'_{02}, and x'_0 by adding 10^{-14}. The NPCR and UACI of the images during the encryption process are displayed in Table 10. The ideal values of the NPCR and UACI were NPCR = 99.6094% and UACI = 33.4635%. It was evident that the NPCR and UACI values closely resembled their theoretical counterparts. In conclusion, Table 10 and Figure 8 demonstrate that the suggested method had significant key sensitivity and can resist known cipher-text attacks.

Table 10. The NPCRs and UACIs between cipher images created with slightly different keys.

Images		Zelda	Einstein	Boat	Ideal
$x'_{01}+10^{-14}$	NPCR	99.5995	99.6323	99.5842	99.6094
	UACI	33.5848	33.4644	33.5536	33.4635
$y'_{02}+10^{-14}$	NPCR	99.6117	99.6307	99.6155	99.6094
	UACI	33.4146	33.5056	33.3489	33.4635
x'_0+10^{-14}	NPCR	99.5544	99.6002	99.5979	99.6094
	UACI	33.4352	33.4323	33.4231	33.4635

4.2.7. Analysis of Differential Attacks

Creating a connection between a plaintext picture and its matching cipher-text image is the fundamental idea behind a differential attack. The sensitivity of a plaintext images determines whether or not the method can withstand differential assaults. Similarly, we encrypted a plaintext image by slightly changing the value of one of the pixels and using the same encryption process. The values of NRCR and UACI of the images before and after the change were calculated, and the results are shown in Table 11. It can be seen that the NRCR and UACI values of the proposed algorithm were still close to the theoretical values with different images. This was because in this work, we used highly sensitive SHA-256 and hyper-chaotic systems to generate a series of parameters. The data compression capabilities in CS and the multi-parameter properties of the 2D LCT were able to improve the effectiveness of the algorithm against differential attacks.

Table 11. The UACIs and NPCRs of various encrypted images.

Images	NPCR	UACI
Ideal	99.6094	33.4635
Zelda	99.6071	33.3957
Einstein	99.5941	33.7801
Boat	99.6300	33.8602
Couple	99.6506	33.9060
Crowded	99.5529	33.0163
Goldhill	99.6452	33.7717
Flinstones	99.6132	33.6949
Bridge	99.5651	32.9519

5. Conclusions

In this study, we illustrated a novel approach to image encryption in 2D LCT domains by utilizing CS, Lorenz's hyper-chaotic system, and chaotic random phase encoding. SHA-256 helps strengthen defenses against specific plaintext attacks. The measurement matrix in CS is connected to the total of an image's pixels, which may further increase the sensitivity to plaintext. CS can also efficiently minimize the size of encrypted images. Two positive Lyapunov exponents and a vast parameter space characterize Lorenz's hyper-chaotic system. It is, therefore, ideal for applications involving image encryption. Additionally, the image is re-encrypted using the 2D LCT based on chaotic random phase masks. The suggested approach has a wide key space and high key sensitivity, and it is especially sensitive to plaintext, according to the simulation findings and security assessments. As such, it is resistant to well-known attacks, such as chosen-plaintext, known-plaintext, and brute-force attacks.

Author Contributions: Conceptualization, Y.-M.L. and D.W.; methodology, Y.-M.L. and D.W.; software, M.J.; validation, D.W.; formal analysis, Y.-M.L. and D.W.; investigation, M.J. and Y.D.; resources, M.J. and Y.D.; data curation, M.J. and Y.D.; writing—original draft preparation, M.J.; writing—review and editing, Y.-M.L. and D.W.; supervision, Y.-M.L. and D.W.; project administration, Y.-M.L. and 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. Matoba, O.; Nomura, T.; Perez-Cabre, E.; Millan, M.S.; Javidi, B. Optical techniques for information security. *Proc. IEEE* **2009**, *97*, 1128–1148. [CrossRef]
2. Wang, Y.; Liu, Z.; Xu, J.; Yan, W. Heterogeneous network representation learning approach for ethereum identity identification. *IEEE Trans. Comput. Soc. Syst.* **2022**, *10*, 890–899. [CrossRef]
3. Zhao, J.; Lv, Y. Output-feedback Robust Tracking Control of Uncertain Systems via Adaptive Learning. *Int. J. Control Autom. Syst.* **2023**, *21*, 1108–1118. [CrossRef]
4. Refregier, P.; Javidi, B. Optical image encryption based on input plane and Fourier plane random encoding. *Opt. Lett.* **1995**, *20*, 767–769. [CrossRef]
5. Situ, G.; Zhang, J. Double random-phase encoding in the Fresnel domain. *Opt. Lett.* **2004**, *29*, 1584–1586. [CrossRef]
6. Chen, J.X.; Zhu, Z.L.; Fu, C.; Yu, H. Optical image encryption scheme using 3-D chaotic map based joint image scrambling and random encoding in gyrator domains. *Opt. Commun.* **2015**, *341*, 263–270. [CrossRef]
7. Wang, X.; Zhao, D. Simultaneous nonlinear encryption of grayscale and color images based on phase-truncated fractional Fourier transform and optical superposition principle. *Appl. Opt.* **2013**, *52*, 6170–6178. [CrossRef]
8. Javidi, B.; Artur Carnicer, E.A. Roadmap on optical security. *J. Opt.* **2016**, *18*, 83001. [CrossRef]
9. Zhou, N.; Wang, Y.; Gong, L. Novel optical image encryption scheme based on fractional Mellin transform. *Opt. Commun.* **2011**, *284*, 3234–3242. [CrossRef]
10. Qin, W.; Peng, X. Asymmetric cryptosystem based on phase-truncated Fourier transforms. *Opt. Lett.* **2010**, *35*, 118–120. [CrossRef]
11. Zhou, N.; Wang, Y.; Wu, J. Image encryption algorithm based on the multi-order discrete fractional Mellin transform. *Opt. Commun.* **2011**, *284*, 5588–5597. [CrossRef]
12. Patra, A.; Saha, A.; Bhattacharya, K. Multiplexing and encryption of images using phase grating and random phase mask. *Opt. Eng.* **2020**, *59*, 33105. [CrossRef]
13. 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]
14. Huang, Z.J.; Cheng, S.; Gong, L.H.; Zhou, N.R. Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform. *Opt. Lasers Eng.* **2020**, *124*, 105821. [CrossRef]
15. 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*, 103. [CrossRef]
16. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284. [CrossRef]
17. Song, C.Y.; Qiao, Y.L.; Zhang, X.Z. An image encryption scheme based on new spatiotemporal chaos. *Optik* **2013**, *124*, 3329–3334. [CrossRef]
18. Wang, X.; Xu, D. A novel image encryption scheme based on Brownian motion and PWLCM chaotic system. *Nonlinear Dyn.* **2014**, *75*, 345–353. [CrossRef]
19. Li, C.; Liu, Y.; Xie, T.; Chen, M.Z.Q. Breaking a novel image encryption scheme based on improved hyperchaotic sequences. *Nonlinear Dyn.* **2013**, *73*, 2083–2089. [CrossRef]
20. Quadri, S.Z.A. Multiple-information security system using spherical wave and chaotic random phase mask encoding. *Opt. Eng.* **2018**, *57*, 93103. [CrossRef]
21. Bechikh, R.; Hermassi, H.; El-Latif, A.A.A.; Rhouma, R.; Belghith, S. Breaking an image encryption scheme based on a spatiotemporal chaotic system. *Signal Process. Image Commun.* **2015**, *39*, 151–158. [CrossRef]
22. 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]
23. Saljoughi, A.S.; Mirvaziri, H. A new method for image encryption by 3D chaotic map. *Pattern Anal. Appl.* **2019**, *22*, 243–257. [CrossRef]
24. Ghazanfaripour, H.; Broumandnia, A. Designing a digital image encryption scheme using chaotic maps with prime modular. *Opt. Laser Technol.* **2020**, *131*, 106339. [CrossRef]
25. 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]
26. 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]
27. 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]
28. Candes, E.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509. [CrossRef]
29. Candes, E.; Wakin, M. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [CrossRef]

30. Huang, X.; Dong, Y.; Ye, G.; Shi, Y. Meaningful image encryption algorithm based on compressive sensing and integer wavelet transform. *Front. Comput. Sci.* **2023**, *17*, 173804. [CrossRef]
31. 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]
32. Chai, X.; Zheng, X.; Gan, Z.; Han, D.; Chen, Y. An image encryption algorithm based on chaotic system and compressive sensing. *Signal Process.* **2018**, *148*, 124–144. [CrossRef]
33. Huang, R.; Rhee, K.H.; Uchida, S. A parallel image encryption method based on compressive sensing. *Multimed. Tools Appl.* **2014**, *72*, 71–93. [CrossRef]
34. 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]
35. Chai, X.; Gan, Z.; Chen, Y.; Zhang, Y. A visually secure image encryption scheme based on compressive sensing. *Signal Process.* **2017**, *134*, 35–51. [CrossRef]
36. Zhang, L.Z.; Zhou, X.; Wang, D.; Li, N.N.; Bai, X.; Wang, Q.H. Multiple-image encryption based on optical scanning holography using orthogonal compressive sensing and random phase mask. *Opt. Eng.* **2020**, *59*, 1. [CrossRef]
37. Hu, G.; Xiao, D.; Wang, Y.; Xiang, T.; Zhou, Q. Securing image information using double random phase encoding and parallel compressive sensing with updated sampling processes. *Opt. Lasers Eng.* **2017**, *98*, 123–133. [CrossRef]
38. 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]
39. Wolf, K.B. *Construction and Properties of Canonical Transforms*; Springer: Berlin/Heidelberg, Germany, 1979; pp. 381–416.
40. Chen, S.S.; Donoho, D.L.; Saunders, M.A. Atomic decomposition by basis pursuit. *Siam Rev.* **2001**, *43*, 129–159. [CrossRef]
41. Pati, Y.; Rezaiifar, R.; Krishnaprasad, P. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–3 November 1993; pp. 40–44.
42. Needell, D.; Tropp, J.A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Commun. ACM* **2010**, *53*, 93–100. [CrossRef]
43. Zhou, N.; Zhang, A.; Wu, J.; Pei, D.; Yang, Y. Novel hybrid image compression–encryption algorithm based on compressive sensing. *Optik* **2014**, *125*, 5075–5080. [CrossRef]
44. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based crytosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [CrossRef]
45. Xingyuan, W.; Junjian, Z.; Guanghui, C. An image encryption algorithm based on ZigZag transform and LL compound chaotic system. *Opt. Laser Technol.* **2019**, *119*, 105581. [CrossRef]
46. Awad, A.; Awad, D. Efficient image chaotic encryption algorithm with no propagation error. *Etri J.* **2010**, *32*, 774–783. [CrossRef]
47. 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]
48. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI Randomness Tests for Image Encryption. *J. Sel. Areas Telecommun.* **2011**, *1*, 31–37.