mathematics

MDPI

Article

A K-SVD Based Compressive Sensing Method for Visual Chaotic Image Encryption

Zizhao Xie [1], Jingru Sun [2,3,*], Yiping Tang [2], Xin Tang [2], Oluyomi Simpson [4] and Yichuang Sun [4]

[1] School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330013, China
[2] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[3] Chongqing Research Institute, Hunan University, Chongqing 401120, China
[4] School of Engineering and Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK
[*] Correspondence: jt_sunjr@hnu.edu.cn

Abstract: The visually secure image encryption scheme is an effective image encryption method, which embeds an encrypted image into a visual image to realize a secure and secret image transfer. This paper proposes a merging compression and encryption chaos image visual encryption scheme. First, a dictionary matrix D is constructed with the plain image by the K-SVD algorithm, which can encrypt the image while sparsing. Second, an improved Zeraoulia-Sprott chaotic map and logistic map are employed to generate three S-Boxes, which are used to complete scrambling, diffusion, and embedding operations. The secret keys of this scheme contain the initial value of the chaotic system and the dictionary matrix D, which significantly increases the key space, plain image correlation, and system security. Simulation shows the proposed image encryption scheme can resist most attacks and, compared with the existing scheme, the proposed scheme has a larger key space, higher plain image correlation, and better image restoration quality, improving image encryption processing efficiency and security.

Keywords: image encryption; compressive sensing; K-SVD; chaos

MSC: 34H10

check for updates

Citation: Xie, Z.; Sun, J.; Tang, Y.; Tang, X.; Simpson, O.; Sun, Y. A K-SVD Based Compressive Sensing Method for Visual Chaotic Image Encryption. *Mathematics* **2023**, *11*, 1658. https://doi.org/10.3390/math11071658

Academic Editors: Jakub Nalepa and Zhanybai T. Zhusubaliyev

Received: 22 February 2023
Revised: 12 March 2023
Accepted: 26 March 2023
Published: 30 March 2023

1. Introduction

The development of information and modern communication technology has made it possible to transmit digital images quickly and conveniently. However, digital images contain much private and secret information, and the protection the security of digital images quickly and effectively has become an urgent problem. Digital images have characteristics of high correlation among pixels, a large amount of data and visualization, etc. Traditional encryption techniques cannot meet their encryption demand. Therefore, digital image encryption (IE) technology has become a hot research topic [1–6].

Chaotic systems have characteristics of pseudo-randomness, uncertainty, and initial value sensitivity, and can be used to obtain pseudorandom sequences [7–10]. In 1989, Mattews [11] first proposed a chaos-based encryption scheme. Then Fridrich proposed a “scrambling-diffusion” IE framework with a two-dimensional chaotic map in 1998 [12]; from then, most digital image encryption methods are based on this framework, and many research results have emerged [13]. There are two main research directions in this field. The first is the application of a new chaotic system. A one dimensional chaotic map is first employed in IE, followed by a multi-dimensional chaotic system. With an in-depth study of chaos theory, hyperchaotic systems [14] and memristor hyperchaotic are also utilized [15–17]. The second direction is the design of the scrambling-diffusion algorithm. Scrambling consists of scrambling the position of each pixel in the picture, and diffusion of spreading the value of each pixel point to other pixel points; a good algorithm for scrambling and diffusion is an important assurance of IE security [18,19]. A large number of different algorithms have been proposed and applied to chaotic IE, such as DNA coding [20,21], S-Box [22], Bit-Level Permutation [23], Bit-plane Rotation [24], Random walk [25,26], etc. However, the scrambling-diffusion framework has two shortcomings. One is that the encrypted image has the same size as the plain image, which provides valuable information for the attacker. The other is the nature of random-like encrypted images, which are easier to recognize as encrypted images and vulnerable to attack.

Compressive sensing can reduce the size of an encrypted image, improve transmission efficiency, solve the first problem mentioned above, and sample, compress, and encrypt the image simultaneously, and has been widely used in image encryption schemes [27]. In a compressive sensing image encryption scheme, the plain image should first be sparse, then a measurement matrix is employed to reduce and encrypt the plain image. Yu et al. proposed constructing the measurement matrix with chaos [28], and Zhou et al. proposed an image compression–encryption scheme to generate a measurement matrix with a logistic map [29].

To solve the second problem, Ye et al. proposed a visual image encryption scheme [30]. In this scheme, the plain image is compressed and encrypted through compressive sensing to obtain a compressed image, then the compressed image is confused and diffused and becomes an encrypted image, and finally the encrypted image is embedded into a carrier image to prevent the attacker from discovering the encrypted image. Based on this scheme, a variety of research results have emerged. For example, Jiang et al. proposed an adaptive embedding algorithm [31], and Yang et al. designed a Generalized Embedding Model [32]. Hua et al. employed an adaptive-thresholding sparsification method, to improve the quality of the reconstructed image [33]. To improve the transmission efficiency of cipher images, a double-image encryption algorithm is designed in [34,35]; furthermore, Liu et al. proposed a multi-image encryption algorithm [36]. These methods improve the efficiency of image encryption and transmission; however, the existing methods of encrypting images with compressive sensing use the measurement matrix as the key, which lacks the correlation of a plain image. The repeated use of the measurement matrix is vulnerable to attack by selected plaintext.

Based on the above analysis, this paper proposes a high plain image correlation measurement matrix compressive sensing image encryption scheme. This scheme includes three steps. First, dictionary D is produced from the plain image with the K-Singular Value Decomposition (K-SVD) algorithm [37]. Based on the singular value decomposition (SVD), the K-SVD algorithm consists of the generalizing of a K-means clustering process, learning an over-complete dictionary from a set of signals. Second, the plain image is sparsed, compressed, and encrypted to a compressed image. Third, three S-Boxes are constructed with the Improved Zeraoulia-Sprott chaotic Map 5 and logistic map and are used to complete the scrambling, diffusion, and embedding operations.

The main contributions of this work are as follows.

First, a compressive sensing algorithm related to plain images is proposed, and a dictionary matrix D is constructed with the plain image by the K-SVD algorithm. D is one of the secret keys and is used to sparse the plain image, then compress the plain image, completing compression and encryption at the same time, which can significantly improve processing efficiency and security.

Second, the secret key of the scheme includes the initial value of the chaotic system and the dictionary of the compressive sensing algorithm, which greatly increases the key space and complexity. Simulations and comparisons show that, compared with the existing schemes, the proposed scheme has larger key space, higher image recovery quality, plain image correlation, flexibility, and a higher security level.

The remainder of the paper is organized as follows. Section II introduces the basic theory of the logistic chaotic system, improved Zeraoulia-Sprott map (IZSM), compressive sensing and S-box. Section III presents the encrypted scheme, the encrypted steps, the encrypted detail and the decryption scheme. Simulation and analysis are performed in Section IV. The last section concludes the paper.

2. Materials and Methods

2.1. Logistic Chaotic System

A logistic map is a simple one-dimensional chaotic map, proposed by American scientist Li and Yorke in 1975 [38], and has been widely used in the field of encrypted communication. The formula for the logistic map is as follows,

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

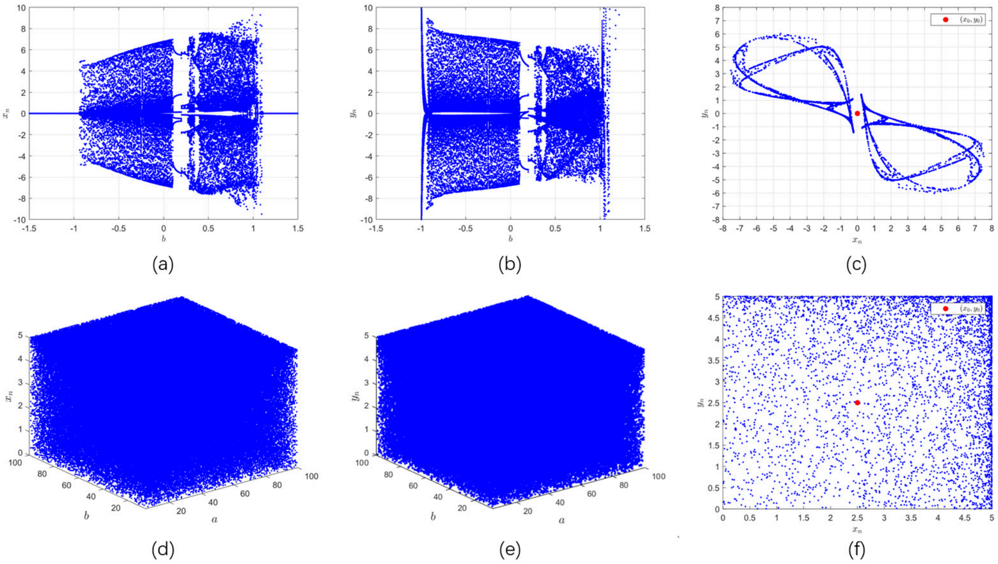
where μ refers to the system parameter whose value range is (0, 4) and x_n refers to the system variable whose value range is (0, 1).

2.2. Improved Zeraoulia-Sprott Map (IZSM)

In this paper, a 2D modular chaotic system(2D-MCS) [39], IZSM, with higher chaos complexity and larger chaotic ranges, is employed, which can effectively solve the problems of narrow chaotic ranges and limited parameter ranges. The employed IZSM is defined as follows,

$$\begin{cases} x_{n+1} = -\hat{a}x_n/\left(1 + y_n^2\right) \; mod \; N \\ y_{n+1} = \left(x_n + \hat{b}y_n\right) \; mod \; N \end{cases}, \tag{2}$$

where $\hat{a}$, $\hat{b}$ are system parameters. The modulus coefficient N is a positive integer. Compared with the limitation of the modular operation in 2D-MCS, IZSM can compact the phase plane significantly. Consequently, the values of the two parameters $\hat{a}$ and $\hat{b}$ in the IZSM are more flexible. As shown in Figure 1, variables x and y in the IZSM can randomly visit the entire regions of the phase plane under all given parameter settings, and the outputs of the IZSM can be randomly distributed on the whole phase plane.



(a) (b) (c)

(d) (e) (f)

Figure 1. Bifurcation diagrams and trajectories of three existing 2D chaotic maps. Panel (**a**–**c**) display the Zeraoulia-Sprott map's bifurcation diagrams under a = 3.8 and b ∈ (−1.5, 1.5), and its trajectory under (a, b) = (3.8, 0.6); (**d**–**f**) bifurcation diagrams and trajectories of the IZSM.

2.3. Compressive Sensing

Compressive sensing can sample, compress, and encrypt the image simultaneously, improving the efficiency of encryption algorithms and the ability to protect the confidentiality of images. The specification of compressive sensing includes three parts, sparsity, uncorrelated observation, and optimum reconstruction of signals.

Given a signal $x \in R^{N\times 1}$, under the action of the orthogonal basis $\Psi \in R^{N\times N}$, the signal x can be sparsely represented as follows,

$$x = \sum_{i=1}^{n} \Psi_i s_i = \Psi_s, \tag{3}$$

where, $s_i = \{s_1, s_2, \ldots s_N\}$ is the coefficient vector of signal x. If the number of non-zero coefficients in vector s is $K \ll N$, i.e., the coefficient vector s_i is K-Sparse, then the signal x is sparse, and can be compressed under the action of the orthogonal basis Ψ. The low-dimensional linear observation value y of length M can be acquired after employing the matrix $\Phi \in R^{M\times N}$of size $M \times N(M \ll N)$ to perform dimensionality reduction observation on the signal. Moreover, y vanishes the correlation of signal x, and the important information of the reconstructed signal x remains,

$$y = \Phi x = \Phi\Psi_s = \Theta s, \tag{4}$$

where $\Theta = \Phi\Psi$ is the sensor matrix, and $y \in R^{M\times 1}$ is the measurement result.

If the original signal x is reconstructed from the observed value y, then the sensor matrix Θ is supposed to satisfy the Restricted Isometry Property (RIP). Some references denote that the measurement matrix Φ does not correlate with the sparse basis Ψ. If the sensor matrix Θ satisfies RIP, the reconstruction problem can be solved by solving the minimum ℓ_0 norm problem of Equation (5),

$$\bar{s} = min||\bar{s}||_{L_1} = min\left|\left|\Phi^T x\right|\right|_{L_1} \tag{5}$$

Finally, by using reverse transformation for $\bar{s}$, the approximate solution vector can be acquired,

$$\bar{x} = \sum_{i=1}^{n} \Psi_i \bar{s}_i = \overline{\Psi}_s. \tag{6}$$

2.4. S-Box

A substitution S-Box is one of the most significant modules in a symmetric key encryption algorithm and has been widely used in combining the complete Latin square with a chaotic system. S-Box is a square matrix that regards numerous bits as input and converts from these bits to the same number of output bits, constructed by the orthogonal matrices generated from the complete Latin square. First, the chaotic system based on the given initial state is used to generate chaotic sequences, and the complete Latin square is generated by a part of the chaotic sequences. As a result, two orthogonal matrices are created. After scrambling the orthogonal matrices based on the chaotic sequences, an S-Box can be constructed. The S-Box makes it easy to change the pixel positions of an image. Analysis results demonstrate that an S-Box has complicated nonlinear properties, can resist different kinds of security attacks, and satisfy the requirement of the security level of the encryption algorithm.

3. Image Encryption Scheme

This section proposes a new K-SVD-based compressive sensing visual chaotic image encryption scheme. There are two stages in the encryption process, encryption and embedding. In the encryption stage, a dictionary is obtained from the plain image by K-SVD, then the plain image is sparsed with the dictionary. The dictionary is one of the secret keys, and the sparse operation is also a kind of encryption operation. The sparsification operation brings better plaintext correlation to the scheme. Next, the sparse image is compressed with a compressive sensing algorithm. The other keys are $x0$ and $y0$, which are the initial value of IZSM and the Logistic map. IZSM generates two pseudorandom sequences to construct S-Boxes, which are used to permute and embed the secret image. In addition, chaos sequences created by a Logistic map are used to diffuse the image. Thus, a plain image is encrypted into an encrypted image. In the embedding stage, the mix coding algorithm embeds the encrypted image into an embedding image with S-Box, and at last the cipher image is obtained.

3.1. OMP

The orthogonal matching pursuit algorithm (OMP) is a regression algorithm. For an underdetermined equation $y = \Theta$s, OMP finds the most approximate solution of s when y (size of $M \times 1$) and Θ (size of $M \times N$) are known. OMP is used in both sparse decomposition and compressive sensing reconstruction in this paper. The solution steps are as follows:

Step 1: Consider each column of the matrix Θ as an atom d,

$$\Theta = (d_1,\ d_1, \ldots, d_n), \tag{7}$$

so, y is the linear combination of atoms in Θ, s is the linear combination coefficient, and s_i measures the contribution of the i-th atom d_i. Next, we calculate the contribution $s_i\langle d_i, y\rangle$ of each atom to y, and select d_i that maximizes this value, where $\langle d_i, y\rangle$ is the absolute value of the result of multiplying the transpose matrix of d_i and y.

Step 2: Set the initial residual $f = y$ and create a new compression matrix $\Theta_{new} = \varnothing$, where $\varnothing$ represents an empty set. Following this, d_i with the largest contribution calculated previously is added into Θ_{new}. Then we subtract d_i from y and the remaining residual is achieved:

$$f = y - \langle d_i, y\rangle d_i. \tag{8}$$

Step 3: Through $\langle d_i, y\rangle$, we can find out d_i with the largest contribution except d_i, and then add d_j to Θ_{new}. Now, the contribution of Θ_{new} to y is required. In order to obtain a new coefficient, OMP will solve the least squares problem:

$$\min||\ \Theta_{new}\omega - y||_2. \tag{9}$$

Then the new coefficient ω is get by

$$\omega = \left(\Theta_{new}^T\Theta_{new}\right)^{-1}\Theta_{new}^T y. \tag{10}$$

The residual is updated to

$$\mathrm{f} = \mathrm{y} - \Theta_{\mathrm{new}}\omega. \tag{11}$$

Step 4: When the residual is less than a certain value, the iteration stops. If the sparsity K is known, it will be iterated K times. Finally, we add the values of ω to the corresponding positions and set the values of other positions as zero, so the reconstructed signal $\hat{s}$ of S can be obtained.

3.2. K-SVD Sparse Dictionary

K-SVD is an algorithm for designing overcomplete dictionaries for sparse representation. Supposing a matrix $Y \in R^{m\times n}$ represents the original signal (image), $Y_{m\times n} = D_{m\times K}X_{K\times n}$, where D is the dictionary matrix with the size of $m \times K$, and X is the corresponding sparse matrix with the size of $K \times n$ which is expected to be as sparse as possible. To deal with this problem, an optimization problem is proposed, which is

$$\min_{D,X}\|\ Y - DY\ \|_F^2 s.t.\forall i, \|\ x_i\ \|_0 \le T_0, \tag{12}$$

or

$$\min_{D,X}\sum\|\ x_i\ \|_0, s.t.\min_{D,X}\|\ Y - DX\ \|_F^2 \le \varepsilon. \tag{13}$$

where $x_i(i = 1,\ 2, \ldots,\ K)$ is the row vector of the sparse matrix X, and $\|\ x_i\ \|_0$ is the zero-order norm representing the number of numbers that are not 0 in the vector. By applying the Lagrangian multiplier method, this problem transforms into a non-constrained optimization problem, which is described as

$$\min_{D,X} \| Y - DX \|_F^2 + \lambda \| x_i \|_1. \tag{14}$$

To make this easy to understand and solve, $\| x_i \|_0$ is replaced by $\| x_i \|_1$.

The general method to optimize X is the OMP algorithm, while the solution for D optimizing can be explained as follows.

Suppose the sparse matrix X is known, the column-wise update of dictionary matrix D can be implemented. The above formula can be converted into

$$\| Y - DX \|_F^2 = \| Y - \sum_{j=1}^{K} d_j x_T^j \|_F^2 = \| (Y - \sum_{j \neq k} d_j x_T^j) - d_k x_T^j \|_F^2 = \| E_k - d_k x_T^j \|_F^2, \tag{15}$$

where d_j is the j-th column vector of D, x_T^k is the k-th row vector of X and the residual $E_k = Y - \sum_{j \neq k} d_j x_T^j$. The optimization problem can be described as

$$\min_{d_k, x_T^k} \| E_k - d_k x_T^k \|_F^2. \tag{16}$$

The problem is transformed into finding the optimal d_k, x_T^k is a least squares question and can be solved by SVD.

To obtain the new x_T^k sparse, the positions in E_k where the corresponding x_T^k is not 0, are extracted, and a new matrix E_k' is rebuilt. As a result, the question is converted into

$$\min_{d_k, x_T^k} \| E_k' - d_k x'^k_T \|_F^2. \tag{17}$$

By employing SVD, we can perform singular value decomposition on E_k',

$$E_k' = U \Sigma V^T. \tag{18}$$

Taking the first column vector of the left singular matrix U as d_k the product of the first-row vector of the right singular matrix as V, and the first singular value Σ as x'^k_T, the corresponding x_T^k can be updated.

3.3. Construction of S-Box

In this paper, we construct the S-Box based on the IZSM, as shown in Figure 2. Three steps are needed.

Step 1: Given the initial keys of IZSM, to generate two sequences x and y, with a length of 10,000, and a value range (0, 1).

Step 2: Normalize x and y sequences to the range (0, 256) and round down, then the normalized sequences x' and y' are obtained.

Step 3: Create a 16 × 16 size matrix, fill the matrix with the first 256 non-repeating values of sequence x', and obtain $Box1$. $Box2$ can be obtained via sequence y' in the same way.

3.4. Coordinate Mapping Algorithm Based on S-Box

In this step, the S-Boxes are employed to perform image coordinate mapping, $Box1$ used for the abscissa and $Box2$ used for the ordinate. Figure 3 shows partial data for $Box1$ and $Box2$. Taking the coordinate (136, 239) as an example, the specific steps of coordinate mapping are as follows. In the 16 × 16 matrix, the 136-th coordinate is located in the 9-th row and 8-th column, so the abscissa corresponds to the coordinate (9, 8) of $Box1$. As shown in Figure 3, the value of this coordinate (9, 8) in $Box1$ is 28, and after adding 1, it is 29. Therefore, the abscissa is mapped from 136 to 29. The same is true for the ordinate. The

239-th coordinate is located in the 15-th row and 15-th column, the coordinate value of *Box*2 (15, 15) is 201, and 202 after adding 1. To sum up, the coordinate (136, 239) becomes (29, 202) after mapping. Because the coordinate range of the encrypted image is [1, 256] and there is no duplicate coordinate, the value range of the S-Box is also [1, 256], and there is no duplicate value, the coordinate mapping is reversible.

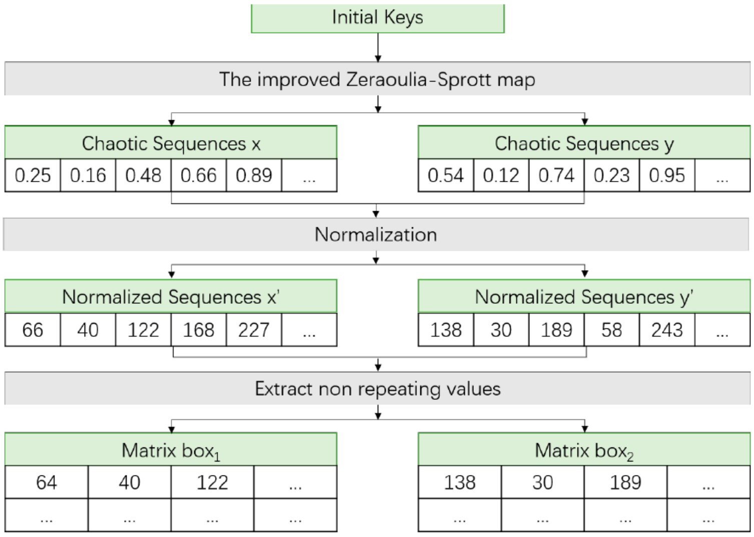


Figure 2. The construction of the S-Box.

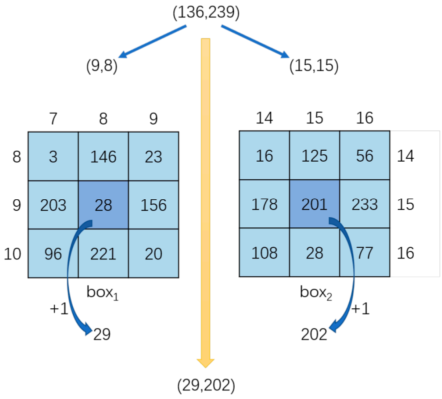


Figure 3. An axis transform example.

3.5. *The Scheme of Image Encryption*

The proposed encryption scheme is given in Figure 4, which includes four stages: compression, scrambling, diffusion, and embedding. The size of the original image P is $M \times N$, the compressed image size is $M \times N$, and the compression ratio is N/M.

3.5.1. Image Compression

Step 1: The dictionary set Dictionary and the SparseMatrix are obtained by the K-SVD algorithm.

$$[\text{Dictionary}, \ \sim] = \text{K} - \text{SVD}(P, \ param), \tag{19}$$

where *param* is the parameter set.

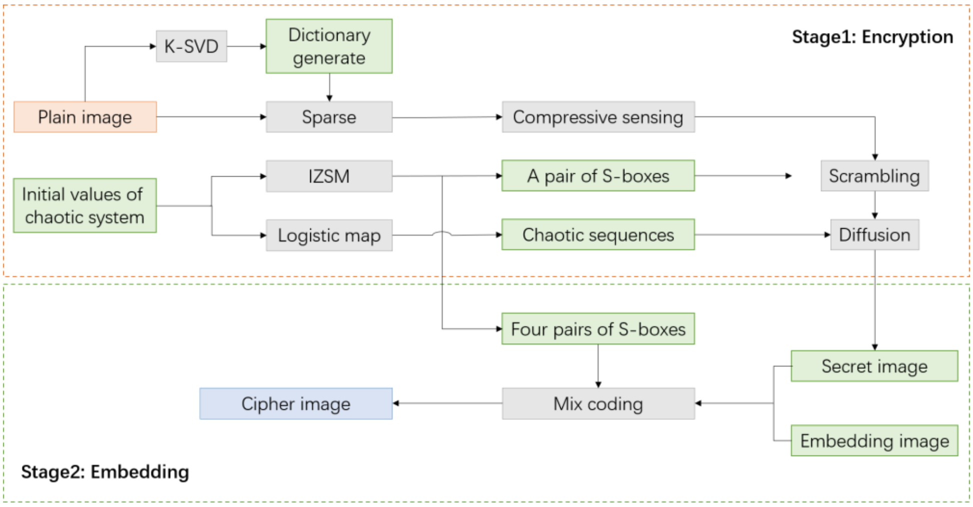


Figure 4. The scheme for image encryption.

$$SparseMatrix = OMP(phi \times Dictionary,\ phi \times P,\ L), \tag{20}$$

where phi is the randomly generated measurement matrix of size $M \times N$ and L is the number of OMP iterations.

Step 2: Construct a random observation matrix PHI of size $M \times N$.

Step 3: Sampling the sparse matrix through the observation matrix PHI to obtain the observation signal Y,

$$Y = PHI \times Dictionary \times SparseMatrix, \tag{21}$$

where the size of Y is $M \times N$.

3.5.2. Image Scrambling

Before scrambling, image Y needs to be supplemented to the size of $N \times N$, and the range of pixel values of each point is required to be (0, 256).

Step 1: Calculate the maximum pixel value $Max1$ and the minimum pixel value $Min1$ in image Y. The pixel value of each point is normalized to (1, 255) with the formula

$$Y(i,\ j) = (Y(i,j) - Min_1 + 1)/(Max_1 - Min_1) \times 255. \tag{22}$$

Step 2: Construct a random matrix Add of size $(N - M) \times N$, normalize each value to (1, 255), and splice the matrix Y with the matrix Add to obtain matrix $Y1$ of size $N \times N$.

Step 3: Two 16 × 16 S-Boxes, $Box1$ and $Box2$ are generated through the IZSM, with the algorithm mentioned in the section "Construction of S-Box".

Step 4: Coordinate mapping is performed on $Y1$ to achieve image scrambling, and $Y2$ is obtained.

3.5.3. Image Diffusion

A logistic chaotic system is utilized to realize image diffusion. First, two $N \times N$ chaotic sequences $S1$ and $S2$ are generated via the Logistic chaotic system, and normalized to the range of [1, 256]. Then, additive modular algorithm is applied to $Y2$ to create $Y3$, called image diffusion. The diffusion process includes forward diffusion and reverse diffusion. The details are as follows,

forward diffusion

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

$$P_i = (256 \times 2 + C_i - S_i - C_{i-1}) mod\, 256 \tag{23}$$

reverse diffusion

$$C_i = (C_{i+1} + P_i + S_i) mod\, 256$$

$$P_i = (256 \times 2 + C_i - S_i - C_{i+1}) mod\, 256 \tag{24}$$

where P is the plain-text, C is the cipher-text, and S is the random sequence.

3.5.4. Image Embedding

Supposing the embedding image is Cr, whose length and width are more than twice that of the encryption image, one pixel of the encryption image can be mapped to four pixels of the embedding image.

Step 1: Four pairs of S-Box (Box_{A1}, Box_{A2}), (Box_{B1}, Box_{B2}), (Box_{C1}, Box_{C2}) and (Box_{D1}, Box_{D2}) are generated through the IZSM.

Step 2: The pixel value of each point in the encryption image is divided into four parts: $Temp1, Temp2, Temp3$, and $Temp4$. $Temp1$ is the integer part. After removing $Temp1$, shift the decimal point right by four digits and take the integer part $Temp2$. After removing $Temp2$, continue to shift the decimal point right by four digits, and take the integer part as $Temp3$. $Temp4$ is equal to $Temp1$.

Step 3: For each pixel part $Tempx$ of each point, its coordinates are mapped from $(x,\ y)$ to (x_1, y_1) through a pair of S-Boxes. Then it is mapped to the coordinates of the embedding image through coordinate transformation,

$$(x_0, y_0) = (2(x_1 - 1) + 1,\ 2(y_1 - 1) + 1). \tag{25}$$

Then the coordinates corresponding to each pixel part are (x_0, y_0), $(x_0 + 1, y_0)$, $(x_0, y_0 + 1)$, $(x_0 + 1, y_0 + 1)$.

Step 4: Each part of the pixel is embedded to the decimal part of the corresponding pixel of the embedding image to achieve ciphertext hiding. The embedded image is C_{r1}.

3.6. Image Decryption Scheme

The decryption process is the opposite of the encryption process and includes four parts: extracting the encryption image from the embedding image, inverse diffusion, inverse scrambling, and compressive sensing recovery. The decryption scheme is shown in Figure 5.

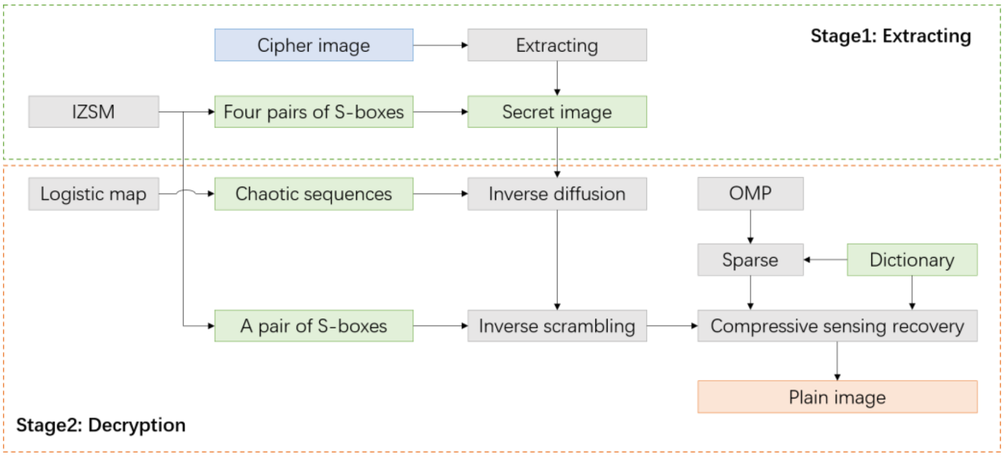


Figure 5. The scheme for image decryption.

Step 1: Extract the decimal part of C_{r1} and record it as C_{r2}.

Step 2: Four pairs of S-Box (Box_{A1}, Box_{A2}), (Box_{B1}, Box_{B2}), (Box_{C1}, Box_{C2}) and (Box_{D1}, Box_{D2}) are generated through the IZSM with the same keys as the encryption process.

Step 3: Create an $N \times N$ zero matrix Y_3 to represent the separated decryption image. For each pixel position (x, y), the mapping coordinate values (x_A, y_A), (x_B, y_B), (x_C, y_C) and (x_D, y_D) are obtained through the above four pairs of S-Box. After the coordinate change, the corresponding coordinates of C_{r2} are obtained $(2(x_A - 1) + 1,\ 2(y_A - 1) + 1)$, $(2(x_B - 1) + 1, 2(y_B - 1) + 1)$, $(2(x_C - 1) + 1,\ 2(y_C - 1) + 1)$ and $(2(x_D - 1) + 1,\ 2(y_D - 1) + 1)$, the four parts of the corresponding pixel can be obtained: $Temp_1$, $Temp_2$, $Temp_3$ and $Temp_4$. In particular, if $Temp_1 = 0$, then $Temp_1 = Temp_4$,

$$Y_3(x, y) = Temp_1 + Temp_2/10^4 + Temp_3/10^8 \tag{26}$$

3.6.1. Inverse Diffusion and Inverse Scrambling

Step 1: Execute the inverse algorithm of reverse diffusion on the encryption image Y_3 via Equation (24), then execute the inverse algorithm of positive diffusion on the image via Equation (23), and the inverse diffusion image Y_2 is generated.

Step 2: Execute the same operation as section "Construction of S-Box" and obtain Box_1 and Box_2.

Step 3: Start from the last coordinate of Y_2 and carry out the following operations consecutively. For each coordinate $(x,\ y)$, obtain the corresponding mapping coordinate (x_1, y_1) through Box_1 and Box_2, exchange the pixel value of the coordinate $(x,\ y)$ and the coordinates (x_1, y_1), and then the image inverse scrambling is finished and Y_1 is obtained.

Step 4: Restore the values of the first M line of Y_1 according to Equation (27),

$$Y_1(i, j) = \frac{Y_1(i, j)}{255} \times (max_1 - min_1) + min_1 - 1. \tag{27}$$

Step 5: Take the first M rows of Y_1 to form a new matrix Y of size $M \times N$.

3.6.2. Compressive Sensing Recovery

The proposed system adopts the OMP algorithm to recover the observed signal, and the required parameters are the observation matrix PHI, the dictionary set $Dictionary$ and the observed signal Y.

4. Simulation Results and Analysis

4.1. Simulation Results

In the following simulation experiments, the experimental environment is windows10 and MATLAB R2016a, CPU: Intel(R) Core (TM) i7-10700 CPU @ 2.90 GHz 2.90 GHz, RAM: 16.0 GB. Four 256 × 256 gray-scale images, Lena, pepper, baboon, and couple are employed as the original images. One 512 × 512 embedded flower image is used for the first two images, and the other two images utilize another embedded mountain image.

The simulation results are exhibited in Figure 6; the pixels of the encrypted images are completely confused, and there is no relevant information. Meanwhile, after the encrypted images and embedded images are mixed, the embedded images almost have no change. Additionally, all decrypted images are identical to the original images.

The average encryption time of four images is 7.69 s, and the embedding time is 0.2 s, reaching the average level of existing work.

4.2. Key Space Analysis

The set of all possible keys in an encryption system is called a key space, denoted as S. Generally, the key space S should be larger than 2^{100} to make brute-force attacks infeasible. In the proposed scheme, the keys are,

(1) the initial values x^0, y^0 of the IZSM and Logistic chaos map.

(2) the dictionary set "*Dictionary*".

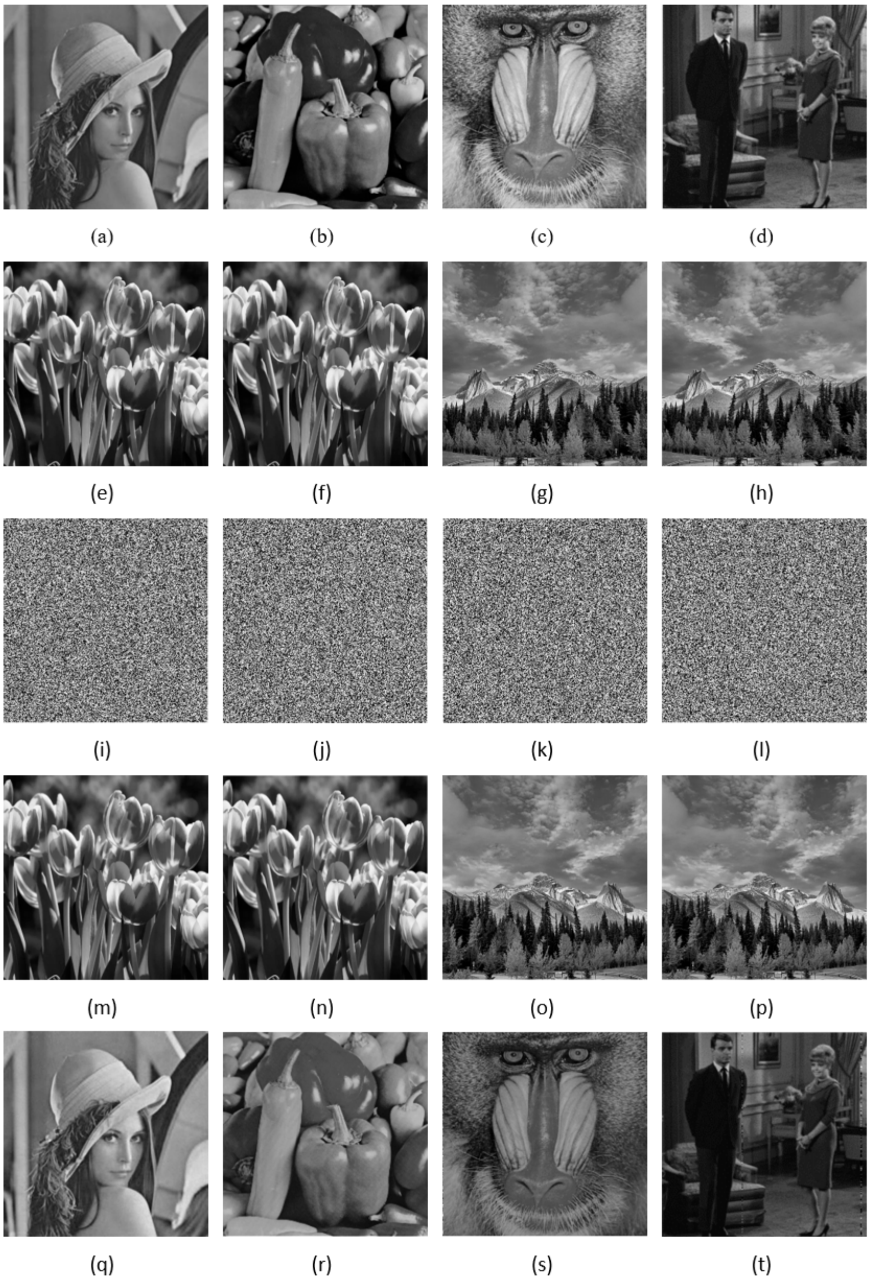
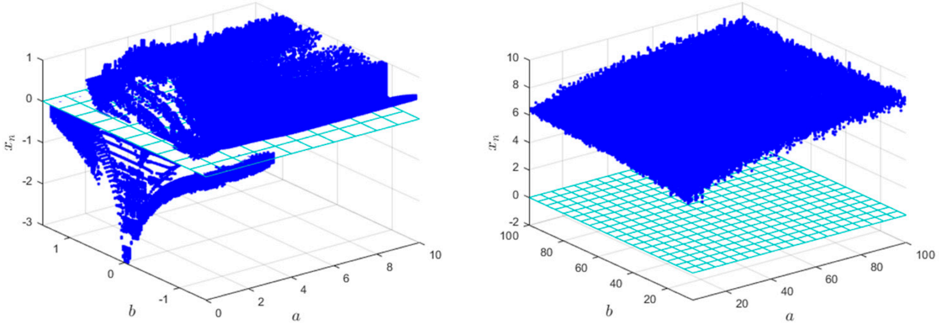


Figure 6. Simulation results. (**a**–**d**) are the original images, (**e**–**h**) are the encrypted images, (**i**–**l**) are the embedded images, (**m**–**p**) are the embedded images with the encrypted images, (**q**–**t**) are the decrypted images.

As mentioned earlier, the IZSM significantly reduces the range limit of the initial values and indirectly increases the difficulty of cracking. Usually, the Lyapunov exponent (LE) is used as an indicator of chaos. As shown in Figure 7, the IZSM can obtain positive LE in a larger parameter range than the original one, which means a larger key space. The key space S in this paper can obtain $S = 10^{166} = 10^{96} \approx 2^{318} \gg 2^{100}$, which is large enough to resist brute-force attack.

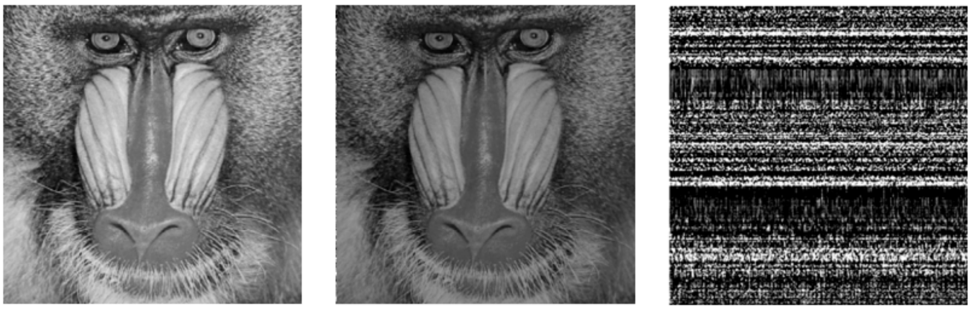


(a) (b)

Figure 7. Key space analysis. (**a**) LE of the Zeraoulia-Sprott map. (**b**) LE of the IZSM.

4.3. Key Security Analysis

Key sensitivity is an important indicator to measure the security of an encryption system. If a system can make the decrypted image irrelevant to the original image after making a tiny change to a key value of the correct keys, it indicates that the key sensitivity of the system is good. The 'baboon' image shown in Figure 8a is used for key sensitivity analysis. The original key is set as $x_0 = 1.1212$, and the slightly modified key is set as $x_0 = 1.12120000000001$. The decrypted image for the correct decryption key is shown in Figure 8b, and the decrypted image for the incorrect decryption key is shown in Figure 8c. The information related to the original image cannot be obtained when the key is changed by only a tiny value of 10^{-14}, which means that the proposed scheme is rather sensitive to the secret keys.



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

Figure 8. Key sensitivity analysis. (**a**) original image; (**b**) decrypted image using the correct decryption key; (**c**) decrypted image using incorrect decryption key.

4.4. Histogram Analysis

A histogram is used to reflect the distribution of the pixel intensity values in the image. For an ideal image encryption system, the histogram of the encrypted image should be as uniform as possible to resist statistical attacks. Figure 9 shows the original images, the encrypted images, and the corresponding histograms. The histogram of the encrypted images is evenly distributed, while the histogram distribution of the original images is uneven. The proposed scheme can effectively withstand statistical attacks.

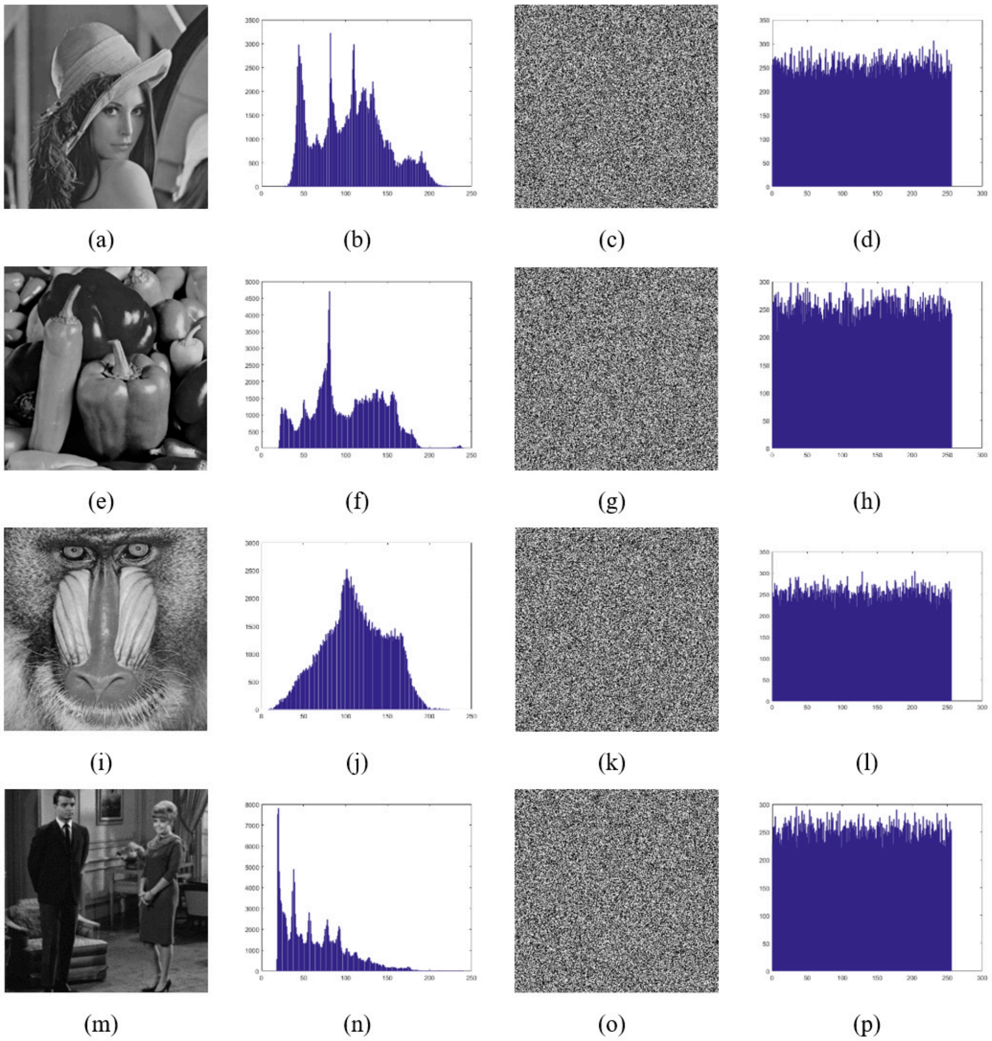
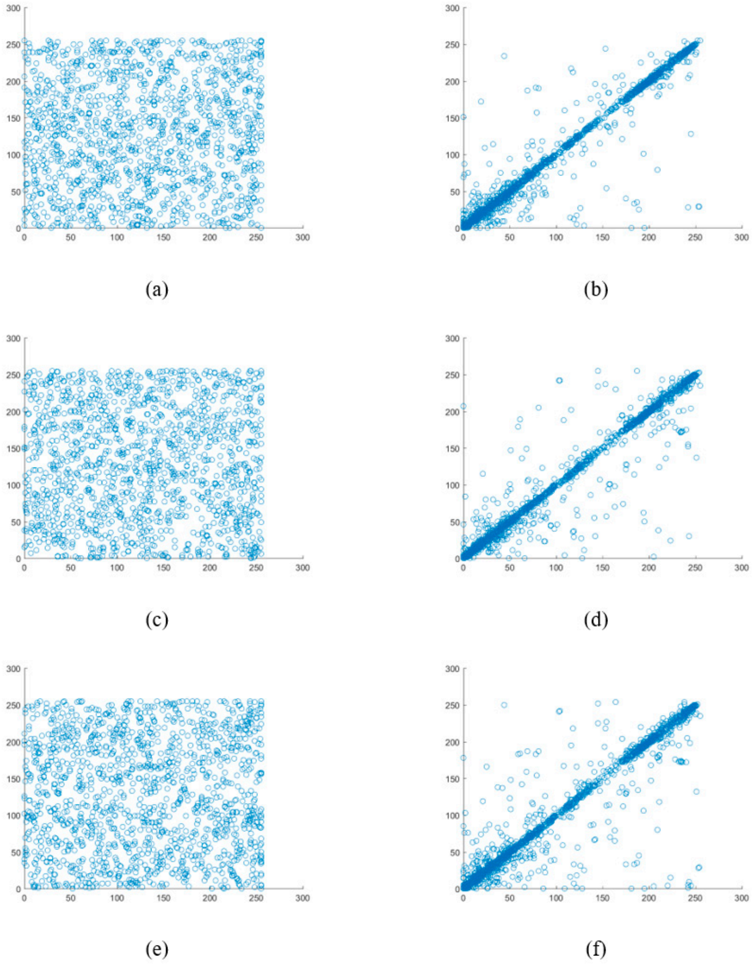


Figure 9. Histogram analysis. (**a**,**e**,**i**,**m**) are the original images, (**b**,**f**,**j**,**n**) are the histograms of the original images, (**c**,**g**,**k**,**o**) are the encrypted images, (**d**,**h**,**l**,**p**) are the histograms of the encrypted images.

4.5. Correlation Analysis

The adjacent pixels of the original images have a high correlation in the horizontal, vertical, and diagonal directions. An ideal image encryption system should sufficiently reduce the relevance between neighboring pixels of the encrypted image to resist statistical attacks. Select 2560 pairs of pixels from the image baboon randomly before and after encryption to draw correlation diagrams. As shown in Figure 10, we can intuitively see that the adjacent pixels of the original images are linearly related, while the adjacent pixels of the encrypted images have a low correlation. Usually, we use correlation coefficient defined by Equation (28) to measure the correlation of adjacent pixels.

$$C_r = \frac{\left(N\sum_{j=l}^{N} x_j y_j - \sum_{j=l}^{N} x_j \sum_{j=l}^{N} y_j\right)}{\left(N\sum_{j=l}^{N} (x_j)^2 - \left(\sum_{j=l}^{N} x_j\right)^2\right)\left(N\sum_{j=l}^{N} (y_j)^2 - \left(\sum_{j=l}^{N} y_j\right)^2\right)} \tag{28}$$



(a) (b) (c) (d) (e) (f)

Figure 10. Correlation analysis. (**a**) Horizontal correlation of encrypted image; (**b**) Horizontal correlation of baboon image; (**c**) Vertical correlation of encrypted image; (**d**) Vertical correlation of baboon image; (**e**) Diagonal correlation of encrypted image; (**f**) Diagonal correlation of baboon image.

The correlation coefficients calculated by Equation (28) are demonstrated in Table 1. The correction coefficients of the original image are very close to 1, while those of the encrypted images are around 0 in all directions, which means that the images after encryption are uncorrelated. Thus, the proposed encrypted algorithm can effectively resist statistical attacks.

Table 1. Correlation coefficient of adjacent pixels.

Image	**Horizontal Correlation Coefficient**	**Vertical Correlation Coefficient**	**Diagonal Correlation Coefficient**
Original image	0.9797	0.9778	0.9615
Encrypted image	−0.0195	0.0191	−0.0051

The number of changing pixel rate (NPCR) and Unified Average Changing Intensity (UACI) are two criteria to quantitatively evaluate the capability of resistance to differential attacks. They are defined by

$$NPCR = \frac{\sum\limits_{i,j} D(i,j)}{W \times H} \times 100\%, \tag{29}$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\% \tag{30}$$

where W and H represent the length and the width of the image, $C_1(i,\ j)$ and $C_2(i,\ j)$ are the pixel values of the two encrypted images, which correspond to the two original images with only 1-bit difference, and $D(i,\ j)$ is defined by

$$D(i,j) = \begin{cases} 0, & \mathrm{C}_1(i,j) = \mathrm{C}_2(i,j) \\ 1, & \mathrm{C}_1(i,j) \neq \mathrm{C}_2(i,j) \end{cases}. \tag{31}$$

The results in Table 2 show that the value of NPCR and UCAI are approaching the ideal theoretical value of 99.61% and 33.46%, indicating the excellent performance of the proposed scheme in resisting the differential attacks.

Table 2. NPCR and UACI of different images.

Image	NPCR	UACI
Lena	99.64%	33.51%
pepper	99.63%	33.44%
baboon	99.64%	33.59%
couple	99.61%	33.56%

4.6. Known-Plaintext Attack Analysis

When attackers know part of the plaintext and the corresponding ciphertext, the general regular pattern of pixel mapping can be obtained via the plaintext ciphertext pair, which is known-plain text attack. A simple linear encryption system is easily cracked by known-plaintext attacks. However, the two-dimensional chaotic system adopted in this paper is nonlinear. At the same time, the one-time encryption method strengthens security. Therefore, the proposed encryption system can resist known-plaintext attacks well.

4.7. Different Compression Ratios

Considering the use of Compressive sensing technology, the reconstruction quality of the image under different compressive ratios is analyzed. Taking two images, Lena and baboon as examples, the decrypted image quality is investigated when the compressive ratios are 8, 4, 2, and 1.33, as shown in Figure 11; when the compressive ratio is 1.33, the decrypted images are of high quality, and are almost the same as the original images. The difference between the decrypted images and the original images is still imperceptible to the human eye when the compressive ratio is 2. When this parameter is set to 4, the decrypted images still have high quality, but a few vertical lines appear. When the compressive ratio increases to 8, the decrypted images become blurred and many vertical lines appear because of the high value of the compressive ratio.

4.8. Sparse Matrix Analysis

The proposed encryption algorithm combines the compressive sensing theory, and the K-SVD algorithm is employed to generate the sparse matrix. Considering that DWT (discrete wavelet transform) is another well-known method for generating sparse matrix, we compare K-SVD and DWT on the quality of the decrypted image, as recorded in Figure 12. It is noticeable that the reconstruction quality of the decrypted image using K-SVD is far better than that using DWT under the same compression ratio. It is also noticeable that the reconstruction quality of the decrypted image using K-SVD is far better than that using DWT under the same compression ratio, but adds encryption and decryption time, which signifies more performance consumption.

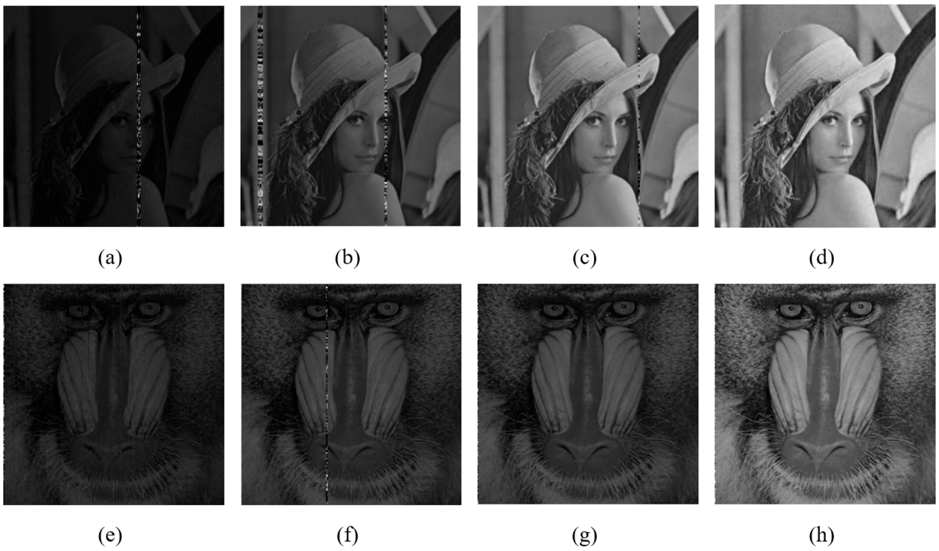


Figure 11. The reconstruction quality of the images of Lena and baboon under different compression ratios. The compression ratios of (**a**) and (**e**) are 8, (**b**) and (**f**) are 4, (**c**,**g**) are 2, (**d**,**h**) are 1.33.

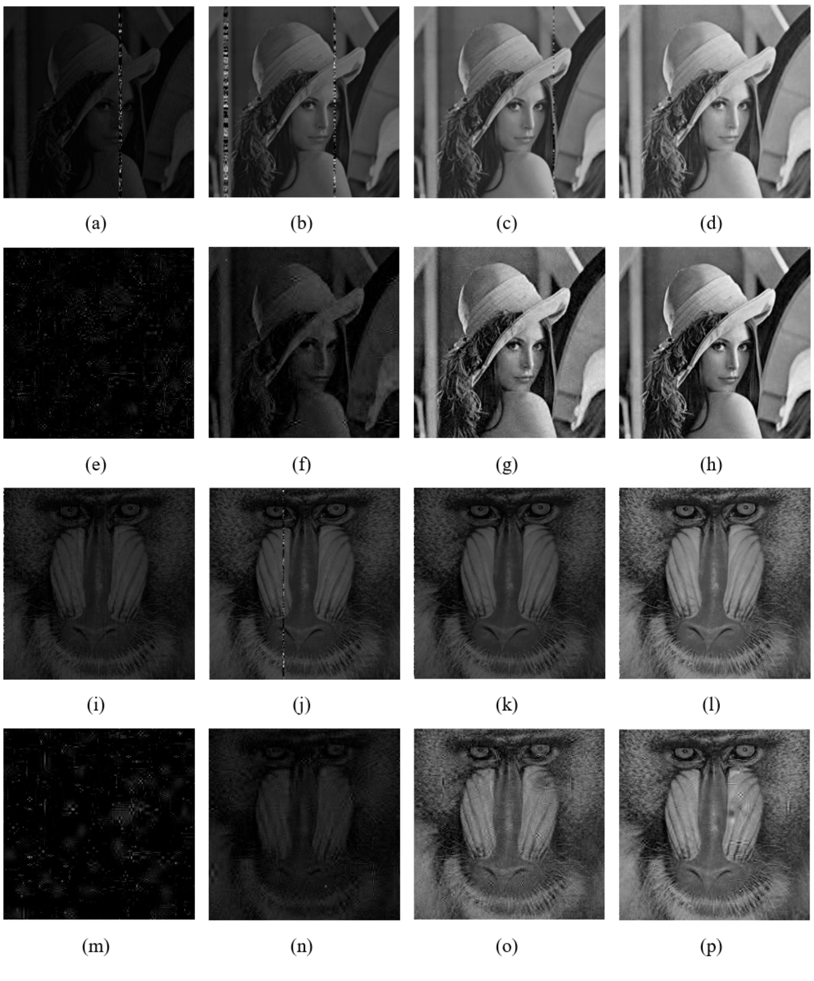


Figure 12. Comparison of DWT and K-SVD. (**a**–**d**,**i**–**l**) are the decrypted images using the K-SVD algorithm, and (**e**–**h**,**m**–**p**) are those using the DWT algorithm. The compression ratios from left to right are 8, 4, 2 and 1.33, respectively.

4.9. Cropping Attack Analysis

During encrypted image transmission, attackers may crop part of the encrypted image, which is called a Cropping Attack. The restoration ability of the decrypted image after cropping is an index to measure the resistance of an algorithm to cropping attacks. Simulations are implemented with the two images, Lena and baboon. Figure 13 shows the decryption images when the embedded images are cropped by 100, 400, and 2500 pixels, respectively.

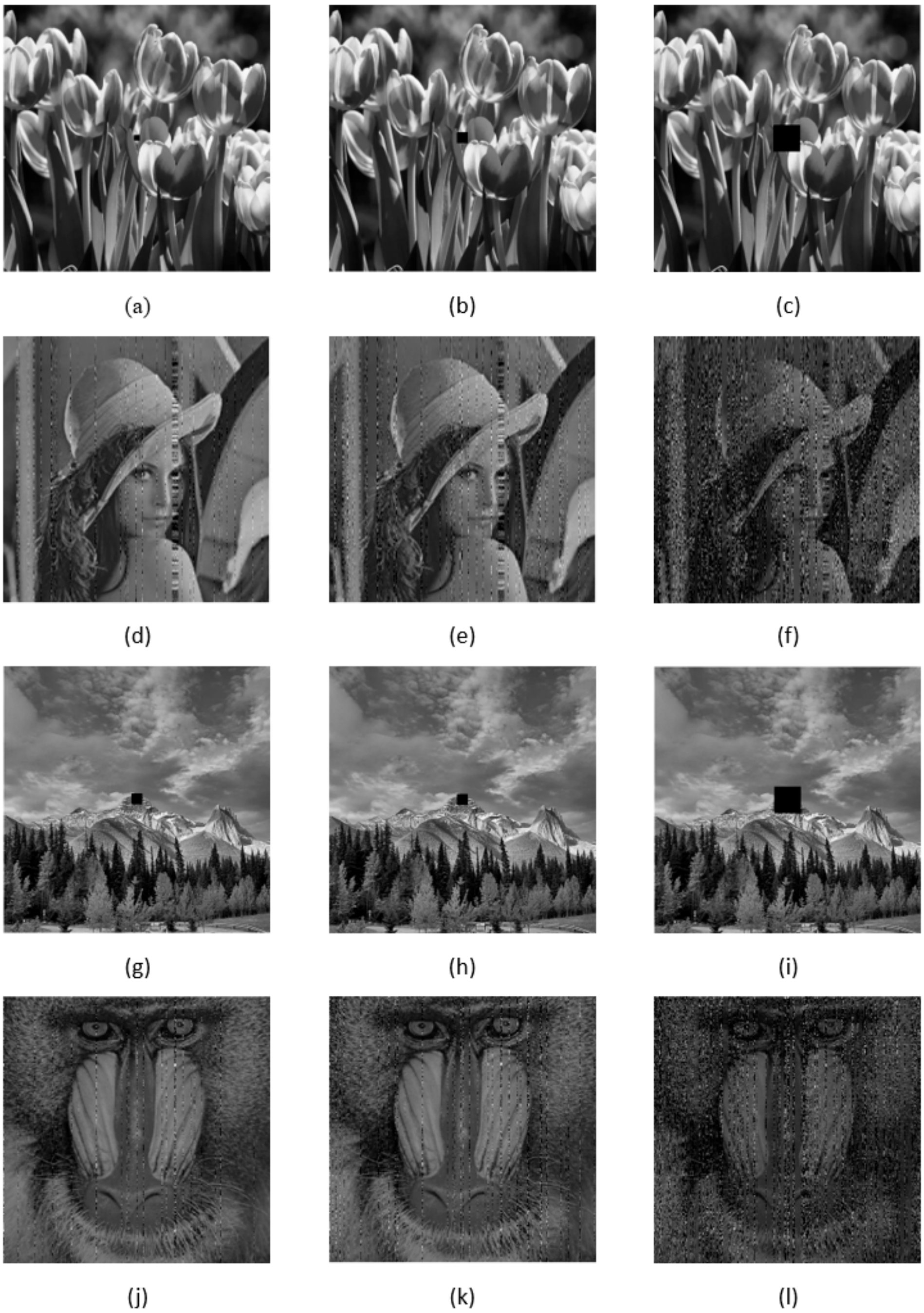


Figure 13. Analysis of images with a portion being cut. (**a**–**c**) The embedded images of flower, cut by 100, 400 and 2500 pixels, respectively. (**d**–**f**) The reconstructed decrypted images of Lena. (**g**–**i**) The embedded images of mountain, cut by 100, 400 and 2500 pixels. respectively. (**j**–**l**) The reconstructed decrypted images of baboon.

As observed, when 100 pixels are cropped, the decrypted images retain most of the information of the original images. When 400 pixels are cropped, the decrypted images lose some details. When 2500 pixels are cropped, many details are lost in the decrypted images, but the content of the images can still be recognized. Thus, the encryption algorithm proposed in this paper has good resistance to cropping attacks.

5. Conclusions

In this paper, we proposed a K-SVD-based compressive sensing chaotic image encryption scheme. The scheme employs the visual secure image encryption structure, through the embedding of the encrypted image into a visual image, increasing the security of image transmission. The key to this scheme includes a dictionary matrix D constructed by K-SVD from the plain image, and the initial value of IZSM chaotic has good plain-image correlation and high quality of image recovery. Three S-Boxes are constructed by the IZSM chaotic sequences and logistic sequences, in order to complete scrambling, diffusion, and embedding. Simulations and comparisons are executed, and the results show that the proposed scheme has larger key space, higher plain image correlation, flexibility, and security level, which can significantly improve the processing image recovery quality and security. Further, the proposed encryption scheme can be optimized in the executing speed, and expanded to achieve color image encryption.

Author Contributions: Conceptualization, Z.X. and J.S.; methodology, Z.X. and Y.T.; software, Y.T.; validation, X.T. and Y.T.; writing—original draft preparation, Z.X., Y.T. and O.S.; writing—review and editing, J.S., Y.S. and O.S.; supervision, Y.S.; project administration, J.S; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by National Science Foundation of China grant number 62171182, 61971185, and the Natural Science Foundation of Hunan Province grant number 2021JJ3014, and the Natural Science Foundation Project of Chongqing, Chongqing Science and Technology Commission grant number CSTB2022NSCQ-M SX0770, and by the Open Fund Project of Key Laboratory in Hunan Universities grant number 20K027.

Data Availability Statement: The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

K-SVD	K-Singular Value Decomposition
IE	digital image encryption
DNA	Deoxy-ribo Nucleic Acid
D	Dictionary
IZSM	Improved Zeraoulia-Sprott Map
2D-MCS	2D modular chaotification system
Ψ	Sparse basis
Θ	Sensor matrix
Φ	Measurement matrix
RIP	Restricted Isometry Property
OMP	orthogonal matching pursuit algorithm
PHI	random observation matrix
LE	Lyapunov exponent

References

1. Lin, H.; Wang, C.; Sun, J.; Zhang, X.; Sun, Y.; Iu, H.H. Memristor-coupled asymmetric neural networks: Bionic modeling, chaotic dynamics analysis and encryption application. *Chaos Solitons Fractals* **2023**, *166*, 112905. [CrossRef]
2. Yu, F.; Shen, H.; Yu, Q.; Kong, X.; Sharma, P.K.; Cai, S. Privacy Protection of Medical Data Based on Multi-Scroll Memristive Hopfield Neural Network. *IEEE Trans. Netw. Sci. Eng.* 2022. [CrossRef]
3. Chinnasamy, P.; Deepalakshmi, P. HCAC-EHR: Hybrid cryptographic access control for secure EHR retrieval in healthcare cloud. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 1–19. [CrossRef]
4. Xiang, T.; Zeng, H.; Chen, B.; Guo, S. BMIF: Privacy-preserving blockchain-based medical image fusion. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 1–23. [CrossRef]
5. Yu, F.; Kong, X.; Mokbel, A.A.M.; Yao, W.; Cai, S. Complex Dynamics, Hardware implementation and image encryption application of multiscroll memeristive hopfield neural network with a novel local active memeristor. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *70*, 326–330. [CrossRef]
6. Chinnasamy, P.; Deepalakshmi, P.; Dutta, A.K.; You, J.; Joshi, G.P. Ciphertext-policy attribute-based encryption for cloud storage: Toward data privacy and authentication in AI-enabled IoT system. *Mathematics* **2021**, *10*, 68. [CrossRef]
7. Yang, T.; Chai, W.W.; Leon, O.C. Cryptography based on chaotic systems. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **1997**, *44*, 469–472. [CrossRef]
8. He, J.; Cui, L.; Sun, J.; Huang, P.; Huang, Y. Chaotic dynamics analysis of double inverted pendulum with large swing angle based on Hamiltonian function. *Nonlinear Dyn.* **2022**, *108*, 4373–4384. [CrossRef]
9. Yu, F.; Shen, H.; Zhang, Z.; Huang, Y.; Cai, S.; Du, S. Dynamics analysis, hardware implementation and engineering applications of novel multi-style attractors in a neural network under electromagnetic radiation. *Chaos Solitons Fractals* **2021**, *152*, 111350. [CrossRef]
10. Wang, M.; Liao, X.; Deng, Y.; Li, Z.; Su, Y.; Zeng, Y. Dynamics, synchronization and circuit implementation of a simple fractional-order chaotic system with hidden attractors. *Chaos Solitons Fractals* **2020**, *130*, 109406. [CrossRef]
11. Matthews, R. On the derivation of a "chaotic" encryption algorithm. *Cryptologia* **1989**, *13*, 29–42. [CrossRef]
12. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284. [CrossRef]
13. Yu, F.; Shen, H.; Zhang, Z.; Huang, Y.; Cai, S.; Du, S. A new multi-scroll Chua's circuit with composite hyperbolic tangent-cubic nonlinearity: Complex dynamics, Hardware implementation and Image encryption application. *Integration* **2021**, *81*, 71–83. [CrossRef]
14. Zeng, J.; Wang, C. A novel hyperchaotic image encryption system based on particle swarm optimization algorithm and cellular automata. *Secur. Commun. Netw.* **2021**, *2021*, 6675565. [CrossRef]
15. Ma, M.; Yang, Y.; Qiu, Z.; Peng, Y.; Sun, Y.; Li, Z.; Wang, M. A locally active discrete memristor model and its application in a hyperchaotic map. *Nonlinear Dyn.* **2022**, *107*, 2935–2949. [CrossRef]
16. Lin, H.; Wang, C.; Sun, Y.; Wang, T. Generating-scroll chaotic attractors from a memristor-based magnetized hopfield neural network. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *70*, 311–315. [CrossRef]
17. Lin, H.; Wang, C.; Yu, F.; Sun, J.; Du, S.; Deng, Z.; Deng, Q. A Review of Chaotic Systems Based on Memristive Hopfield Neural Networks. *Mathematics* **2023**, *11*, 1369. [CrossRef]
18. Zhu, Y.; Wang, C.; Sun, J.; Yu, F. A Chaotic Image Encryption Method Based on the Artificial Fish Swarms Algorithm and the DNA Coding. *Mathematics* **2023**, *11*, 767. [CrossRef]
19. Wang, X.; Zhang, X.; Gao, M.; Tian, Y.; Wang, C.; Iu, H.H.-C. A Color Image Encryption Algorithm Based on Hash Table, Hilbert Curve and Hyper-Chaotic Synchronization. *Mathematics* **2023**, *11*, 567. [CrossRef]
20. Sun, J.; Peng, M.; Liu, F.; Tang, C. Protecting compressive ghost imaging with hyperchaotic system and DNA encoding. *Complexity* **2020**, *2020*, 8815315. [CrossRef]
21. Cheng, S.; Sun, J.; Xu, C. A color image encryption scheme based on a hybrid cascaded chaotic system. *Int. J. Bifurc. Chaos* **2021**, *31*, 2150125. [CrossRef]
22. Hua, Z.; Li, J.; Chen, Y.; Yi, S. Design and application of an S-box using complete Latin square. *Nonlinear Dyn.* **2021**, *104*, 807–825. [CrossRef]
23. 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]
24. Xu, C.; Sun, J.; Wang, C. An image encryption algorithm based on random walk and hyperchaotic systems. *Int. J. Bifurc. Chaos* **2020**, *30*, 2050060. [CrossRef]
25. Zhou, Y.; Li, C.; Li, W.; Li, H.; Feng, W.; Qian, K. Image encryption algorithm with circle index table scrambling and partition diffusion. *Nonlinear Dyn.* **2021**, *103*, 2043–2061. [CrossRef]
26. Yu, F.; Zhang, Z.; Shen, H.; Huang, Y.; Cai, S.; Du, S. FPGA implementation and image encryption application of a new PRNG based on a memristive Hopfield neural network with a special activation gradient. *Chin. Phys. B* **2022**, *31*, 020505. [CrossRef]
27. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [CrossRef]
28. Yu, L.; Barbot, J.P.; Zheng, G.; Sun, H. Compressive sensing with chaotic sequence. *IEEE Signal Process. Lett.* **2010**, *17*, 731–734. [CrossRef]
29. 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]

30. Ye, G.; Pan, C.; Dong, Y.; Shi, Y.; Huang, X. Image encryption and hiding algorithm based on compressive sensing and random numbers insertion. *Signal Process.* **2020**, *172*, 107563. [CrossRef]
31. Jiang, D.; Liu, L.; Zhu, L.; Wang, X.; Rong, X.; Chai, H. Adaptive embedding: A novel meaningful image encryption scheme based on parallel compressive sensing and slant transform. *Signal Process.* **2021**, *188*, 108220. [CrossRef]
32. Yang, Y.G.; Wang, B.P.; Pei, S.K.; Zhou, Y.H.; Shi, W.M.; Liao, X. Using M-ary decomposition and virtual bits for visually meaningful image encryption. *Inf. Sci.* **2021**, *580*, 174–201. [CrossRef]
33. Hua, Z.; Zhang, K.; Li, Y.; Zhou, Y. Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing. *Signal Process.* **2021**, *183*, 107998. [CrossRef]
34. Huang, W.; Jiang, D.; An, Y.; Liu, L.; Wang, X. A novel double-image encryption algorithm based on Rossler hyperchaotic system and compressive sensing. *IEEE Access* **2021**, *9*, 41704–41716. [CrossRef]
35. Ye, G.; Liu, M.; Wu, M. Double image encryption algorithm based on compressive sensing and elliptic curve. *Alex. Eng. J.* **2022**, *61*, 6785–6795. [CrossRef]
36. Ye, G.; Pan, C.; Dong, Y.; Jiao, K.; Huang, X. A novel multi-image visually meaningful encryption algorithm based on compressive sensing and Schur decomposition. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4071. [CrossRef]
37. Aharon, M.; Elad, M.; Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **2006**, *54*, 4311–4322. [CrossRef]
38. Li, T.Y.; Yorke, J.A. Period three implies chaos. *Amer. Math* **1975**, *82*, 975. [CrossRef]
39. Hua, Z.; Zhang, Y.; Zhou, Y. Two-dimensional modular chaotification system for improving chaos complexity. *IEEE Trans. Signal Process.* **2020**, *68*, 1937–1949. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.