mathematics

MDPI

Article

Technique for Enhancing the Chaotic Characteristics of Chaotic Maps Using Delayed Coupling and Its Application in Image Encryption

Shuiyuan Huang [1], Gengsheng Deng [2,*], Lingfeng Liu [3,*] and Xiangjun Li [1,3]

[1] School of Mathematics and Computer Science, Nanchang University, Nanchang 330031, China; huangshuiyuan@ncu.edu.cn (S.H.); lixiangjun@ncu.edu.cn (X.L.)
[2] Network Center, Nanchang University, Nanchang 330031, China
[3] School of Software, Nanchang University, Nanchang 330031, China
* Correspondence: dgsheng@ncu.edu.cn (G.D.); lfliu@ncu.edu.cn (L.L.)

Abstract: Chaotic systems are widely used in many scientific fields for their dynamic characteristics. This study proposes a new delayed coupling method, which not only disturbs the control coefficient in chaotic maps but also affects their function structure, such that using this improved method will produce chaotic maps with better effect. The numerical simulation results prove that the delayed coupling method can greatly improve the chaotic characteristics of chaotic maps. Furthermore, an image encryption algorithm based on the delayed coupling Logistic map is proposed. Several numerical simulations indicate that the image encryption algorithm has a high level of security, and can compete with other encryption algorithms.

Keywords: chaos; coupling; delayed state; image encryption

MSC: 65P20; 94A08

Citation: Huang, S.; Deng, G.; Liu, L.; Li, X. Technique for Enhancing the Chaotic Characteristics of Chaotic Maps Using Delayed Coupling and Its Application in Image Encryption. *Mathematics* **2023**, *11*, 3295. https://doi.org/10.3390/math11153295

Academic Editor: Antanas Cenys

Received: 20 June 2023
Revised: 12 July 2023
Accepted: 24 July 2023
Published: 26 July 2023

1. Introduction

Chaos is a complex physical phenomenon. Chaotic maps are highly sensitive to initial conditions; control coefficients; and long-term unpredictable, aperiodic, and complex dynamics. All these characteristics make chaotic maps popular in many natural and social scientific fields, including computer communication, biomedical engineering, experimental physics, electronic information, economics, and cryptography.

Since the first chaotic system, namely, the Lorenz system, many different types of chaotic systems have been proposed and studied. One-dimensional (1D) chaotic maps have the simplest structure among chaotic systems. Such chaotic systems are always easy to realize; however, their dynamical complexity is generally relatively lower than that of other chaotic systems. Representative 1D chaotic systems include Logistic maps [1], Chebyshev maps [2], and Tent maps [3]. Supposing that the Euclidean dimensions of a chaotic system are greater than 2, in such a case, it constitutes a high-dimensional (HD) chaotic system, examples of which include the Lorenz system [4], Henon maps [5], Cat maps [6], and the Chen system [7]. Generally, these chaotic systems will always have a higher dynamical complexity; however, their implementation efficiency is relatively lower. In addition, hyperchaotic systems, with at least two positive Lyapunov exponents, causes these types of systems exhibit rich, dynamic behaviors. Hyperchaotic systems are always constructed by coupling multiple chaotic systems, and examples include fractional-order hyperchaotic systems [8], Lorenz hyperchaotic systems [9], and Qi hyperchaotic systems [10].

When using chaotic systems in cryptography, dynamical complexity and implementation efficiency should be considered. Thus, HD (i.e., more than three dimensions) chaotic systems are not recommended due to their low implementation efficiency. Furthermore,

existing low-dimensional chaotic systems have the disadvantages of a small parameter space, simple structure, and low complexity, causing them to not be secure enough for cryptographic uses. Therefore, an ideal method is to enhance the dynamical complexity of low-dimensional chaotic systems.

To date, many methods have been developed to improve the chaotic properties of low-dimensional chaotic maps [11–18]. Among them, Tang, J. et al. [11] constructed a new one-dimensional cosine and Logistic composite chaotic map with complex chaotic behavior. Xiang, H.Y. et al. [12] proposed a method for enhancing chaotic dynamics by destroying the state space using sinusoidal functions as feedback functions. Liu, L.F. et al. [13] proposed a simple perturbation method to reduce the dynamic degradation of numerical chaotic maps. Liu, B.C. et al. [14] designed an improved method, which was made up of feedback control with linear function and parameter disturbance, and state variables were applied to control the next state and system parameters. Most of these methods enhanced the chaotic properties by introducing external sources of chaos. Still, all of them were limited in that the dynamics of the external chaotic sources directly affected the properties of the improved chaotic system. Therefore, in recent years, a feasible new method has been proposed by researchers, namely the delayed coupling method, which can be applied to all chaotic systems. Liu, L.F. et al. [15] first introduced the properties of delayed states and redesigned the control parameters of chaotic systems in combination with linear functions, eventually leading to improved dynamics of chaotic systems. Liu, L.F. et al. [16] used a bi-coupled method to ameliorate the dynamic characteristics of chaotic maps. Li, S. et al. [17] used the delayed and linear coupling of a one-dimensional Logistic map to achieve enhanced chaotic dynamical properties. Tang, J.Y. et al. [18] employed the delayed and linear coupling method to a 1D Logistic map, which exhibited improved chaotic performance. However. the delayed states were only used to perturb the control coefficient of chaotic maps, which would not affect the nonlinear function of the chaotic map. In the present study, the delayed states are used to affect the structure of the chaotic function, which will have a better effect on chaotification. The advantages of this method can be described as follows.

(1) The delayed coupling method can greatly enhance the chaotic characteristics.
(2) This method is universal and can be applied to different chaotic maps.
(3) The delayed coupling method is simple and low cost.

The rest of this paper is organized as follows. The basic framework of the chaotic coupling model is introduced in Section 2. In Section 3, a delay-coupled Logistic model is initially proposed and analyzed. Section 4 advances a novel image encryption algorithm for the delay-coupled logistic model. Several experiments are presented to prove the security of this encryption algorithm. Finally, Section 5 summarizes this research.

2. Novel Delayed Coupled Chaotic Models

Consider the following chaotic maps:

$$x_{i+1} = f(x_i, p) \tag{1}$$

where x_i denotes the state variable, p is the control coefficient, and f is the nonlinear function. Generally, when the coefficient p is taken to be in a specific range, map f becomes chaotic.

For these reasons, the structures of 1D chaotic maps are relatively simple, and the complexity is low. Therefore, their characteristics need to be improved. A novel coupling method is proposed in this study. The following are the coupling subsystems:

$$y_{i+1} = g(y_i, q) \tag{2}$$

Similarly, y_i denotes the state variable, q is the control coefficient, and g is the nonlinear function. Thus, the delay-coupled chaotic model can be described as follows:

$$\begin{cases} x_{i+1} = f(h_1(y_i), p(x_{i-1})) \\ y_{i+1} = g(h_2(x_i), q(y_{i-1})) \end{cases} \tag{3}$$

In this model, h_1 and h_2 are the state control functions, and p and q are the coefficient control functions. From Equation (3), it can be seen that state x_{i+1} is controlled by y_i, and y_{i+1} is controlled by x_i. This method introduces delayed states, and also couples the two one-dimensional maps into a new two-dimensional model. Meanwhile, the coefficient control function is used to generate coefficient variables controlled by their delayed states. As a result, the coefficient variables change during the iterative process, causing the obtained sequences to be unstable, and thus increasing the complexity of the chaotic sequences. To ensure that the coupled model is chaotic, two conditions should be satisfied.

(1) Given that chaotic functions f and g can be different, the value range of the variables x and y will be different. Thus, the state control functions h_1 and h_2 should cause the coupled states y_i and x_i to fall into the value ranges of the chaotic functions f and g, respectively.
(2) Functions f and g will be chaotic when the coefficients are in the region of the chaotic parameters. Thus, the range of coefficient control functions p and q should be in the region of chaotic parameters to ensure that the functions are chaotic.

This delayed coupling method is suitable for all chaotic maps, whether they are identical or different. In the next sections, two Logistic maps will be coupled in order to demonstrate the effectiveness of the method. Naturally, this method could be extended to the coupling of multiple chaotic systems. The delay coupling model can be extended to N models, as follows:

$$\begin{cases} x_{i+1}^{(1)} = f_1\left(h_1\left(x_i^{(2)}, x_i^{(3)}, \cdots, x_i^{(N)}\right), p_1\left(x_{i-1}^{(1)}\right)\right) \\ x_{i+1}^{(2)} = f_2\left(h_2\left(x_i^{(1)}, x_i^{(3)}, \cdots, x_i^{(N)}\right), p_2\left(x_{i-1}^{(2)}\right)\right) \\ \cdots \\ x_{i+1}^{(N)} = f_N\left(h_N\left(x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(N-1)}\right), p_N\left(x_{i-1}^{(N)}\right)\right) \end{cases} \tag{4}$$

where f_1, f_2, …, f_N are the chaotic maps; h_1, h_2, …, h_N are the state control functions; and p_1, p_2, …, p_N are the coefficient control functions. It is also necessary for the state control functions and coefficient control functions to satisfy Conditions 1 and 2. The implementation costs of the delayed coupled chaotic model will increase with increasing value of N. Therefore, in this study, we only set $N = 2$ with comprehensive consideration.

3. The Delayed Coupled Logistic Chaotic Model and Its Characteristics

3.1. The Delayed Coupled Logistic Chaotic Model

The Logistic map may be the most widely used 1D chaotic map in a number of different scientific fields. The Logistic map can be expressed with the following formula:

$$x_{i+1} = F(x_i, a) = ax_i(1 - x_i) \tag{5}$$

where a is a control parameter. Generally, the range of a is (3.5699, 4), this map is chaotic. Next, the delayed coupling states can be selected, and the coefficient control functions can be described as follows:

$$\begin{cases} h_1(y_i) = y_i \cdot (1 - y_i) \\ h_2(x_i) = x_i \cdot (1 - x_i) \end{cases} \tag{6}$$

$$\begin{cases} p_1(x_{i-1}) = a + (4 - a) \cdot min(x_{i-1}, x_i) \\ p_2(y_{i-1}) = a + (4 - a) \cdot min(y_{i-1}, y_i) \end{cases} \tag{7}$$

Here, h_1 and h_2 are the state control functions, and p_1 and p_2 are the coefficient control functions. Finally, the new delayed coupling model can be described as follows:

$$\begin{cases} x_{i+1} = (a + (4 - a) \cdot min(x_{i-1}, x_i) \cdot (y_i) \cdot (1 - y_i)) \\ y_{i+1} = (a + (4 - a) \cdot min(y_{i-1}, y_i) \cdot (x_i) \cdot (1 - x_i)) \end{cases} \tag{8}$$

In this model, we know that the following state is controlled by the current state of another subsystem, that is, state x_{i+1} is controlled by the function y_i, and y_{i+1} is controlled by the function x_i. The coefficient control functions are used to define the coefficient variable, which is controlled by its delayed state. In the next section, the common approach of analyzing the efficiency of this will be discussed.

3.2. Characteristic Analysis

In this section, several numerical simulations are presented to show the effectiveness of the delayed coupling method. The parameters are selected as $a = 3.99$, $x_1 = 0.32$ for Equation (5), $a = 3.99$, $x_1 = 0.32$, $x_2 = 0.36$, $y_1 = 0.423$, and $y_2 = 0.436$ for Equation (8).

3.2.1. Trajectories and Phase Diagrams

An ideal chaotic system will have a random trajectory without any regular structural characteristics, and will also possess a good ergodicity in the phase space. The trajectories of the improved map and the primitive map are depicted in Figure 1. Figure 1 shows that after 800 iterations, the trajectory of the improved chaotic map still remains unstructured and irregular. Figure 2a,b present the phase diagrams of Equation (8) in the x- and y-dimensions, respectively. Figure 2c presents the phase diagram produced by Equation (5). From Figure 2, it can be seen that the delayed coupling method can completely disrupt the phase space of the original Logistic map. Both trajectories and phase diagrams illustrate that the delayed coupled Logistic map has better randomness and ergodicity.

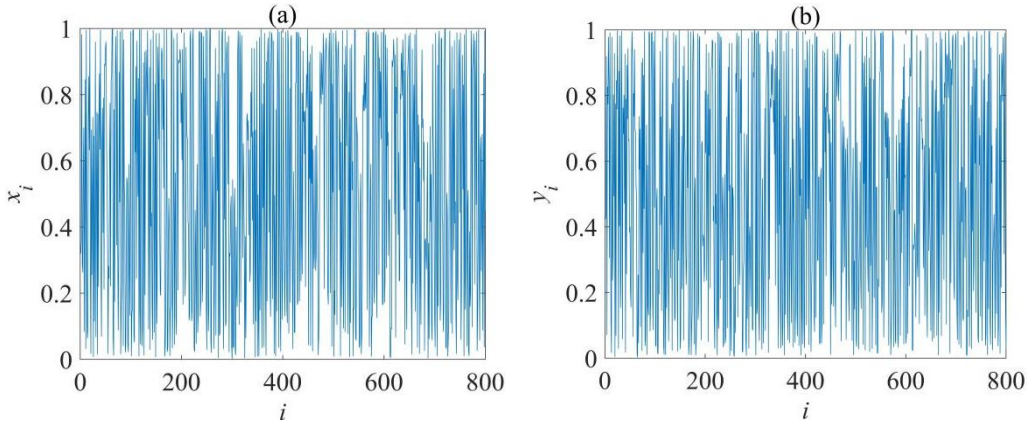

Figure 1. Trajectories of Equation (8): (**a**) x-dimension; (**b**) y-dimension.

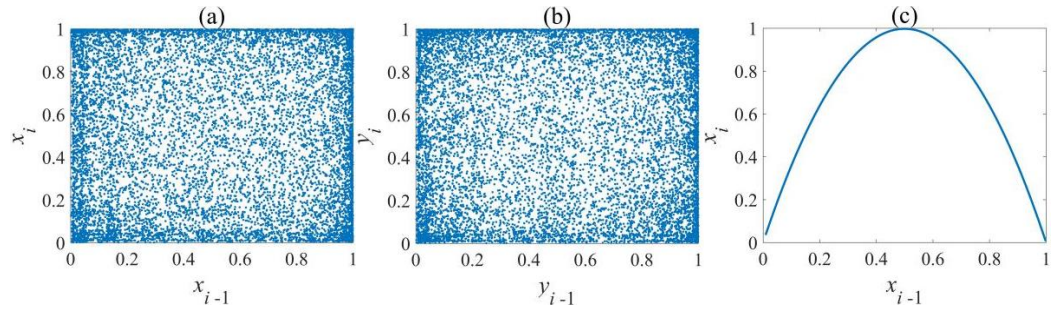

Figure 2. Phase diagrams: (**a**) x-dimension of Equation (8); (**b**) y-dimension of Equation (8); (**c**) Equation (5).

3.2.2. Bifurcation Diagram Analysis

The bifurcation diagram intuitively reflects the values of the control parameters, which causes the system to be chaotic. Figure 3 plots the bifurcation diagrams of the results of Equations (5) and (8). When varying the value of a, the value of the sequence traverses the interval (0, 1), indicating that the system is bounded. Figure 3a implies that when the

control parameter is in the interval (3.5699, 4), the initial Logistic system becomes chaotic. Meanwhile, from Figure 3b,c, it can be observed that Equation (8) will become chaotic, since the control parameter is greater than 3, which indicates that Equation (8) has a larger chaotic parameter space. Furthermore, Equations (5) and (8) will not be chaotic for some specific parameters, although the parameters are located in the chaotic area. These can be referred to as period windows, which is a common phenomenon in chaos theory.

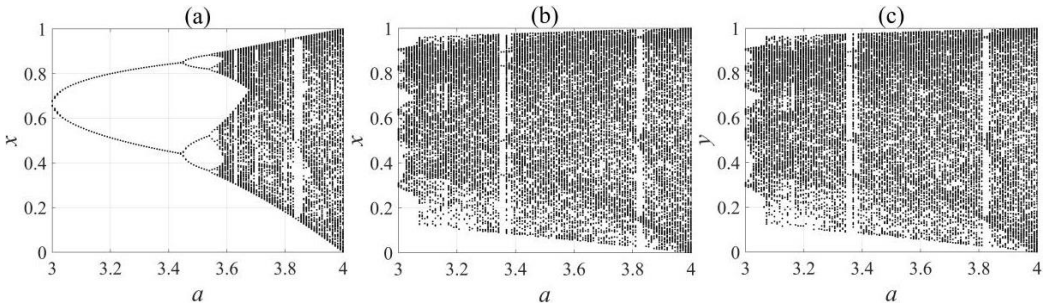

Figure 3. Bifurcation diagrams: (**a**) Equation (5); (**b**) $x-$dimension of Equation (8); (**c**) $y-$dimension of Equation (8).

3.2.3. Approximate Entropy Analysis

Approximate entropy (ApEn) is often used to characterize chaotic maps, with larger values indicating that the system expresses more complex dynamic behavior. Figure 4 shows the ApEn value curves produced by the x-dimensional variables of Equation (5) and Equation (8), respectively. The ApEn values generated by the y-dimensional chaotic sequence are similar to that of the x-dimensional variable, which is omitted here to avoid redundancy. From Figure 4, it can be seen that for almost all parameters, Equation (8) will always generate greater ApEn values than Equation (5), which proves that the improved chaotic map is characterized by greater complexity in this case. Furthermore, when the parameter a is in the interval (3.8, 3.84), the ApEn curve of Equation (8) will exhibit a rapid decrease due to the period window. Similar results can be found in the ApEn curve of Equation (5), as well.

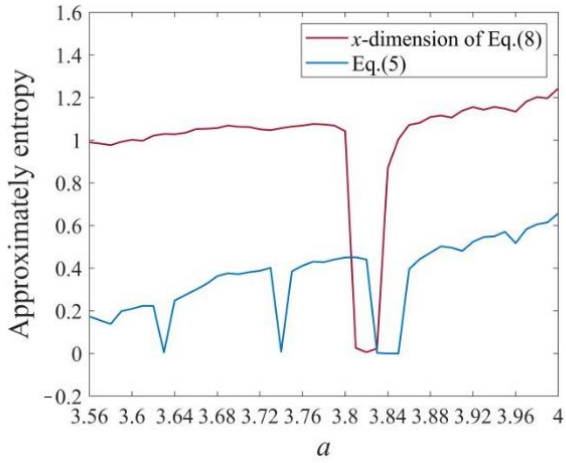

Figure 4. ApEn analysis of Equations (5) and (8).

3.2.4. Permutation Entropy Analysis

Permutation entropy (PE) introduces the idea of permutation to evaluate the complexity of reconstructed sub-sequences. Similar to ApEn, it is an important index for expressing the complexity of time series. As in the case of the ApEn analysis, the PE values of Equations (5) and (8) were calculated, and the results are provided in Figure 5. From Figure 5, it can be found that the PE values of Equation (8) are always greater than those of Equation (5), except when the control parameter a falls into the period window, which

indicates that the delayed coupling Logistic map successfully enhanced the complexity of original Logistic map in this case.

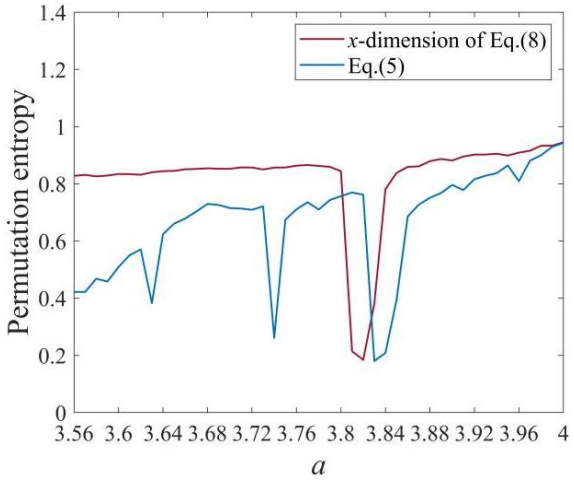

Figure 5. PE analysis of Equations (5) and (8).

3.2.5. Lyapunov Exponent Analysis

The motion characteristics of a nonlinear system can be reflected by the Lyapunov exponent (LE). A system is regarded as chaotic if it has at least one positive LE. In this test, we compared the LE values generated using Equations (5) and (8) to judge whether the delayed coupling method can enhance the chaotic properties. The results of the LE curves are shown in Figure 6. Figure 6 shows that the improved Logistic map is chaotic (LE > 0), since parameter a is greater than 3, meaning that it can access chaos faster. Furthermore, except for the parameters in the period window of Equation (8), the LE of Equation (8) will be greater than that of Equation (5) for all other parameters, thus proving that the delayed coupling method can effectively improve chaotic behavior.

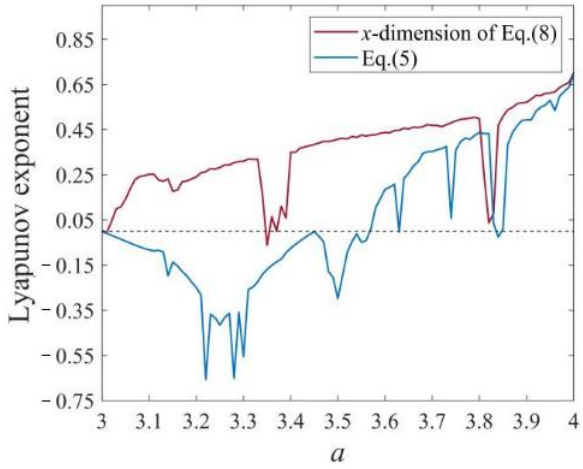

Figure 6. LE analysis of Equations (5) and (8).

3.2.6. Auto-Correlation Analysis

The auto-correlation function is used to describe the degree of correlation of the given sequences at different times, thus further validating their randomness. Figure 7 provides the auto-correlation function of Equation (8). The results indicate that the auto-correlation function has a peak value at zero, while with increasing interval value, the auto-correlation value decreases rapidly to 0. Therefore, the sequence generated by Equation (8) exhibits little correlation, indicating that good randomness has been achieved in this case.

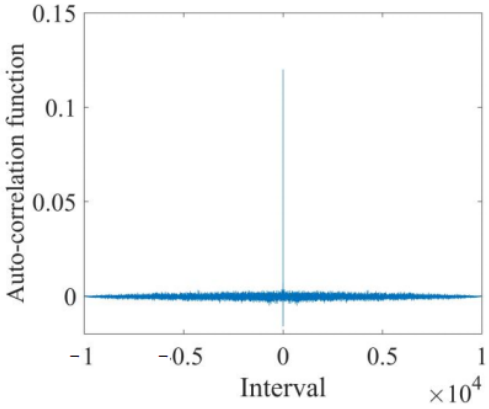

Figure 7. Auto-correlation analysis of Equation (8).

3.2.7. Sensitivity Analysis

Chaotic systems are sensitive to both initial conditions and parameters, which means that small changes in the initial values and the parameters can produce completely different chaotic sequences. Figure 8 depicts the sensitivity analysis of x-dimensional variables in Equation (8) for different initial values and parameters, respectively. From Figure 8, it can be found that with slight changes in the initial values and parameters, the trajectories separate completely after only 20 iterations, thus demonstrating that Equation (8) is highly sensitive to the initial values and control parameters.

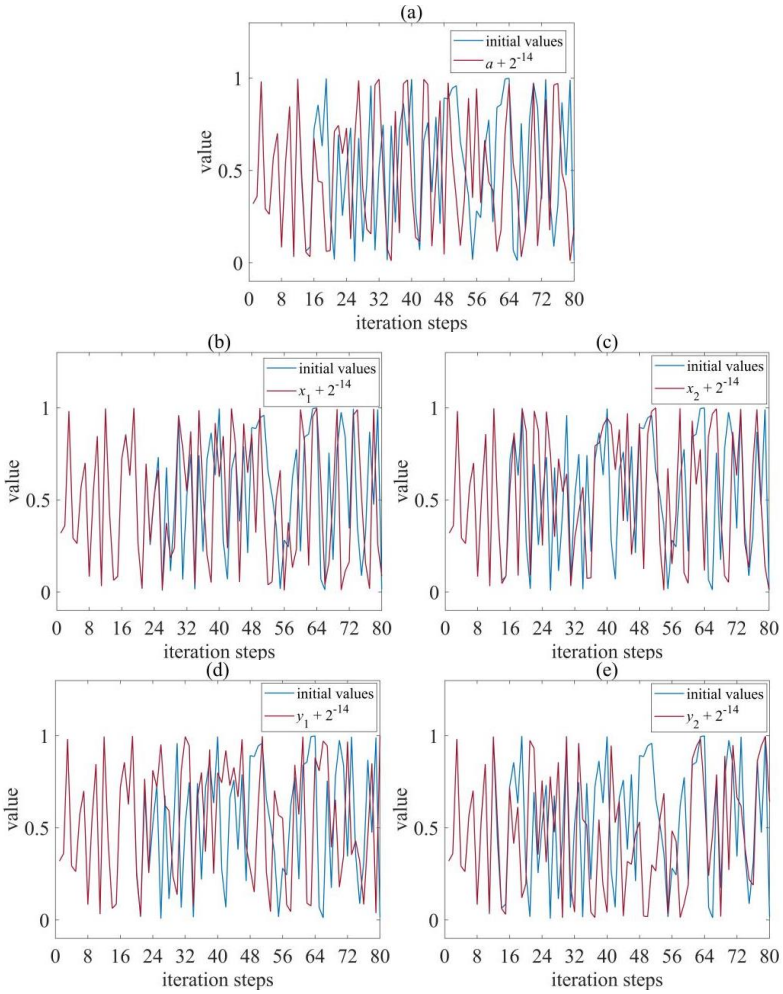

Figure 8. Sensitivity analysis of x-dimensional variable of Equation (8): (**a**) a, (**b**) x_1, (**c**) x_2, (**d**) y_1, (**e**) y_2.

4. Image Encryption Algorithm Based on the New Model

To verify the validity of the chaos model, we propose a new encryption algorithm. This algorithm can not only be used to demonstrate the practicability of a chaotic system, it can also provide a simple concept for performing image encryption. This method utilizes the random sequences generated by the new delayed coupling system to scramble the pixel matrices. The randomness of the chaotic sequence can enhance the encryption effect, while the numerical analysis also confirms the encryption effect with this novel chaotic system.

4.1. Algorithm in Image Encryption and Decryption

4.1.1. Shuffling Algorithm

A new shuffle algorithm is proposed to sort and transform the pixel matrix so as to make the pixel matrix disordered and achieve image encryption. In this experiment, only one column transformation was performed. The simple encryption algorithm obtained better results because of its ability to improve chaotic mapping. The algorithm can be summarized as described below.

First, an original plain image L is read, the size of L is assumed to be $M \times N$, and its gray pixel matrix J_{ij} is obtained.

$$J_{ij} = \begin{pmatrix} j_{11} & j_{12} & j_{13} & \cdots & j_{1N} \\ j_{21} & j_{22} & j_{23} & \cdots & j_{2N} \\ j_{31} & j_{32} & j_{33} & \cdots & j_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ j_{M1} & j_{M2} & j_{M3} & \cdots & j_{MN} \end{pmatrix} \tag{9}$$

Second, the chaotic sequences are scrambled. A matrix K_{ij} of size $M \times N$ is transformed by the chaotic sequence generated by Equation (8) after $M \times N$ iterations. The fractional value a and the maximum value n of the average value of each column i of the matrix K_{ij} is obtained.

$$\begin{cases} a = mean(K) \\ n = ceil(a(i)) \\ n1 = M - n \\ a = a - floor(a) \end{cases} \tag{10}$$

where $mean(\cdot)$ is used to find the average value of each column of the matrix. $ceil(\cdot)$ is a logarithmic up integer. $floor(\cdot)$ rounds the value down. These three functions are all included in MATLAB 2019 software.

For each column I of the chaotic sequence matrix K_{ij} and the pixel matrix J_{ij}, i ranges from 1 to M. The following transformation is performed.

$$\text{If } rem(i, 2) = 0$$

$$\begin{cases} circshift(J(:, i), n) \\ circshift(K(:, i), i) \end{cases} \tag{11}$$

$$\text{else} \begin{cases} circshift(J(:, i), i + n1) \\ circshift(K(:, i), n) \end{cases} \tag{12}$$

where J(:,i) and K(:,i) are the i-th column of J_{ij} and K_{ij}, respectively. $rem(\cdot)$ is a complementary function to judge odd and even numbers. $circshift(\cdot)$ is a circular translation function that represents the shifting up of the i-th column. These two functions are also implemented in the MATLAB 2019 software package. For the different columns of matrix J_{ij} and K_{ij}, after different times of cycling, the matrices J'_{ij} and $K1_{ij}$ will be obtained.

Finally, the chaotic matrix J'_{ij} is transformed into a sequence $\{J\}$ with the same size as the chaotic sequence $\{K\}$. After that, $K1_{ij}$ is converted into sequence $\{K1\}$, and $\{K1\}$ is sorted in ascending order to obtain $\{K1'\}$. Moreover, the sequence $\{J\}$ is sorted depending on the

order of the sequence $\{K1'\}$. $\{J'\}$ is obtained and retransformed into a matrix $J1_{ij}$, which is the final shuffle matrix. The main process is carried out as follows:

$$J'_{ij} = \begin{pmatrix} j'_{11} & j'_{12} & j'_{13} & \cdots & j'_{1N} \\ j'_{21} & j'_{22} & j'_{23} & \cdots & j'_{2N} \\ j'_{31} & j'_{32} & j'_{33} & \cdots & j'_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ j'_{M1} & j'_{M2} & j'_{M3} & \cdots & j'_{MN} \end{pmatrix} K1_{ij} = \begin{pmatrix} k'_{11} & k'_{12} & k'_{13} & \cdots & k'_{1N} \\ k'_{21} & k'_{22} & k'_{23} & \cdots & k'_{2N} \\ k'_{31} & k'_{32} & k\prime_{33} & \cdots & k\prime_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k'_{M1} & k'_{M2} & k\prime_{M3} & \cdots & k\prime_{MN} \end{pmatrix}$$

$$\Downarrow$$

$$\{J\} = \{j_1, j_2, j_3, \cdots, j_{M\times N}\}(j_1 = j'_{11}, j_2 = j'_{12}, j_3 = j'_{13}, \cdots, j_{M\times N} = j'_{MN})$$

$$\{K1\} = \{k1_1, k1_2, k1_3, \cdots, k1_{M\times N}\}(k1_1 = k'_{11}, k1_2 = k'_{12}, k1_3 = k'_{13}, \cdots, k1_{M\times N} = k'_{MN})$$

$$\Downarrow \quad \begin{matrix} \{Kl\}\ Ascending \\ \{J\}\ sorting \end{matrix}$$

$$\{K1'\} = \{k1'_{l1}, k1'_{l2}, k1'_{l3}, \cdots, k1'_{lM\times N}\}(k1'_{l1} < k1'_{l2} < k1'_{l3} < \cdots < k1'_{lM\times N})$$

$$\{J'\} = \{j'_1, j'_2, j'_3, \cdots, j'_{M\times N}\}(j'_{l1} = j_1, j'_{l2} = j_2, j'_{l3} = j_3, \cdots, j'_{lM\times N} = j_{M\times N})$$

$$\Downarrow$$

$$J1_{ij} = \begin{pmatrix} j1_{11} & j1_{12} & j1_{13} & \cdots & j1_{1N} \\ j1_{21} & j1_{22} & j1_{23} & \cdots & j1_{2N} \\ j1_{31} & j1_{32} & j1_{33} & \cdots & j1_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ j1_{M1} & j1_{M2} & j1_{M3} & \cdots & j1_{MN} \end{pmatrix} (j1_{11=}j'_1, j1_{12=}j'_2, j1_{13=}j'_3, \cdots, j1_{MN} = j\prime_{M\times N})$$

4.1.2. Image Encryption and Decryption Algorithm

The image encryption algorithm can be described by the following steps.

Step 1: Read the plain image L with size of $M \times N$, and let J_{ij} be its pixel matrix.

Step 2: Obtain the average values of the matrix J_{ij} and the fractional part b.

$$b = mean(J) - floor(mean(J)) \tag{13}$$

Step 3: Set initial parameters and values (a, x_1, x_2, y_1, and y_2) for Equation (8), The fractional part b of the average value generated in Step 2 is added to the initial values. This method causes the final chaos sequence to be affected by the primary image.

$$\begin{cases} x'_1 = (x_1 \cdot b) \cdot (mod\ 1) \\ x'_2 = (x_2 \cdot b) \cdot (mod\ 1) \\ y'_1 = (y_1 \cdot b) \cdot (mod\ 1) \\ y'_2 = (y_2 \cdot b) \cdot (mod\ 1) \end{cases} \tag{14}$$

Step 4: The chaotic sequence $\{K\}$ is obtained by iterating the improved system $M \times N$ times. Then, $\{K\}$ is transformed into K_{ij} with a size of $M \times N$.

Step 5: By using J_{ij} and K_{ij} matrices, two new matrices $J1_{ij}$ and $K1_{ij}$ are obtained using the scrambling algorithm described in Section 4.1.1.

Step 6: Matrices K_{ij} and $K1_{ij}$ are normalized to (0, 255), then they are turned into matrices G_{ij} and H_{ij}, respectively. The normalized chaotic matrix H_{ij} is sorted in descending order to form a new matrix $H1_{ij}$.

$$\begin{cases} G_{ij} = \left(uint8\left(255 * k_{ij}\right)\right) \\ H_{ij} = \left(uint8\left(255 * k1_{ij}\right)\right) \end{cases}$$
$$\Downarrow$$
$$H1_{ij} = \begin{pmatrix} h1_{11} & h1_{12} & h1_{13} & \cdots & h1_{1N} \\ h1_{21} & h1_{22} & h1_{23} & \cdots & h1_{2N} \\ h1_{31} & h1_{32} & h1_{33} & \cdots & h1_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h1_{M1} & h1_{M2} & h1_{M3} & \cdots & h1_{MN} \end{pmatrix} \left(h1_{1l} > h1_{2l} > h1_{3l} > \cdots > h1_{Ml}\right)$$
$$(l = 1, 2, 3, 4, \cdots, N)$$

Step 7: The ordered matrix $H1_{ij}$ and the scrambled matrix $J1_{ij}$ in Step 5 are used to obtain the initial encryption matrix E'_{ij}.

$$E' = H1 \bigoplus J1 \tag{15}$$

Step 8: The final encryption matrix E is obtained using the XOR between the initial encryption matrix E' obtained in Step 7 and the chaotic matrix G generated in Step 6.

$$E = E' \bigoplus G \tag{16}$$

Figure 9 shows the procedure of this encryption algorithm. Similarly, the decryption process can be explained as follows, and the flowchart of the decryption procedure is presented in Figure 10.

Step 1: Read the encrypted image E, and obtain its gray pixel matrix E_{ij}.

Step 2: Use the initial values (x_1', x_2', y_1', y_2', and a) to make the improved Logistic map (Equation (8)) generate new chaotic sequences $\{K\}$. Then, the chaotic sequence matrix K_{ij} is normalized to (0, 255) to obtain the matrix G_{ij}. The specific conversion method is as follows:

$$\{K\} = \{k_1, k_2, k_3, \cdots, k_{M\times N}\}$$
$$\Downarrow$$
$$K_{ij} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & \cdots & k_{1N} \\ k_{21} & k_{22} & k_{23} & \cdots & k_{2N} \\ k_{31} & k_{32} & k_{33} & \cdots & k_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{M1} & k_{M2} & k_{M3} & \cdots & k_{MN} \end{pmatrix}$$
$$\Downarrow \qquad g_{ij} = unit8\left(255 * k_{ij}\right)$$
$$G_{ij} = \begin{pmatrix} g_{11} & g_{12} & g_{13} & \cdots & g_{1N} \\ g_{21} & g_{22} & g_{23} & \cdots & g_{2N} \\ g_{31} & g_{32} & g_{33} & \cdots & g_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{M1} & g_{M2} & g_{M3} & \cdots & g_{MN} \end{pmatrix}$$

Step 3: K_{ij} employs a shuffling algorithm to obtain chaotic matrix $K1_{ij}$. Then, matrix $K1_{ij}$ is normalized to (0, 255) to become matrix H_{ij}. Finally, matrix H_{ij} is arranged in descending order to obtain a new matrix $H1_{ij.}$

$$K1_{ij} = \begin{pmatrix} k1_{11} & k1_{12} & k1_{13} & \cdots & k1_{1N} \\ k1_{21} & k1_{22} & k1_{23} & \cdots & k1_{2N} \\ k1_{31} & k1_{32} & k1_{33} & \cdots & k1_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k1_{M1} & k1_{M2} & k1_{M3} & \cdots & k1_{MN} \end{pmatrix}$$

$$\Downarrow \quad h_{ij} = unit8(255 * k1_{ij})$$

$$H_{ij} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1N} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2N} \\ h_{31} & h_{32} & h_{33} & \cdots & h_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{M1} & h_{M2} & h_{M3} & \cdots & h_{MN} \end{pmatrix}$$

$$\Downarrow \quad H_{ij}\ descending$$

$$H1_{ij} = \begin{pmatrix} h1_{11} & h1_{12} & h1_{13} & \cdots & h1_{1N} \\ h1_{21} & h1_{22} & h1_{23} & \cdots & h1_{2N} \\ h1_{31} & h1_{32} & h1_{33} & \cdots & h1_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h1_{M1} & h1_{M2} & h1_{M3} & \cdots & h1_{MN} \end{pmatrix} (h1_{1l} > h1_{2l} > h1_{3l} > \cdots > h1_{Ml})$$

$$(l = 1, 2, 3, 4, \cdots, N)$$

Step 4: Matrix E' is obtained using the XOR operation of matrices G and $H1$, and perturbation matrix $J1$ is then obtained using the XOR operation of E' and encrypted matrix E.

$$E' = G \bigoplus H1 \tag{17}$$

$$J1 = E' \bigoplus E \tag{18}$$

Step 5: Chaos matrix $K1_{ij}$ and pixel matrix $J1_{ij}$ are reordered according to the shuffle algorithm, and a new pixel matrix J'_{ij} is obtained.

$$J1_{ij} = \begin{pmatrix} j1_{11} & j1_{12} & j1_{13} & \cdots & j1_{1N} \\ j1_{21} & j1_{22} & j1_{23} & \cdots & j1_{2N} \\ j1_{31} & j1_{32} & j1_{33} & \cdots & j1_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ j1_{M1} & j1_{M2} & j1_{M3} & \cdots & j1_{MN} \end{pmatrix} K1_{ij} = \begin{pmatrix} k1_{11} & k1_{12} & k1_{13} & \cdots & k1_{1N} \\ k1_{21} & k1_{22} & k1_{23} & \cdots & k1_{2N} \\ k1_{31} & k1_{32} & k1_{33} & \cdots & k1_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k1_{M1} & k1_{M2} & k1_{M3} & \cdots & k1_{MN} \end{pmatrix}$$

$$\Downarrow$$

$$\{J\} = \{j_1, j_2, j_3, \cdots, j_{M\times N}\}(j_1 = j1_{11}, j_2 = j1_{12}, j_3 = j1_{13}, \cdots, j_{M\times N} = j1_{MN})$$

$$\{K1\} = \{k1_1, k1_2, k1_3, \cdots, k1_{M\times N}\}(k1_1 = k1_{11}, k1_2 = k1_{12}, k1_3 = k1_{13}, \cdots, k1_{M\times N} = k1_{MN})$$

$$\Downarrow \quad \{Kl\}\ Ascending$$

$$\{J\}\ sorting$$

$$\{K1'\} = \{k1'_{l1}, k1'_{l2}, k1'_{l3}, \cdots, k1'_{lM\times N}\}(k1'_{l1} < k1'_{l2} < k1'_{l3} < \cdots < k1'_{lM\times N})$$

$$\{J'\} = \{j'_1, j'_2, j'_3, \cdots, j'_{M\times N}\}(j'_{l1} = j_1, j'_{l2} = j_2, j'_{l3} = j_3, \cdots, j'_{lM\times N} = j_{M\times N})$$

$$\Downarrow$$

$$J'_{ij} = \begin{pmatrix} j'_{11} & j'_{12} & j'_{13} & \cdots & j'_{1N} \\ j'_{21} & j'_{22} & j'_{23} & \cdots & j'_{2N} \\ j'_{31} & j'_{32} & j'_{33} & \cdots & j'_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ j'_{M1} & j'_{M2} & j'_{M3} & \cdots & j'_{MN} \end{pmatrix} (j'_{11} = j'_1, j'_{12} = j'_2, j'_{13} = j'_3, \cdots, j'_{MN} = j'_{M\times N})$$

Step 6: When moving the matrix inversely, the moving times are different. The number of moves is as follows:

$$\text{If } rem(i, 2) = 0$$

$$\begin{cases} circshift(J'(:,i), n1) \\ circshift(K(:,i), M-i) \end{cases} \tag{19}$$

$$\text{else} \begin{cases} circshift(J'(:,i), M-i-n1) \\ circshift(K(:,i), n1) \end{cases} \tag{20}$$

The column of $K1_{ij}$ and J'_{ij} circularly moves up at the corresponding times, respectively, which can invert the encrypted pixel matrices. K is the chaotic matrix transformed from the chaotic sequence $\{K\}$, and J' is a pixel matrix sorted inversely.

Step 7: From Step 6, an inverse disturbance matrix J_{ij} is obtained, which is the decrypted gray pixel matrix.

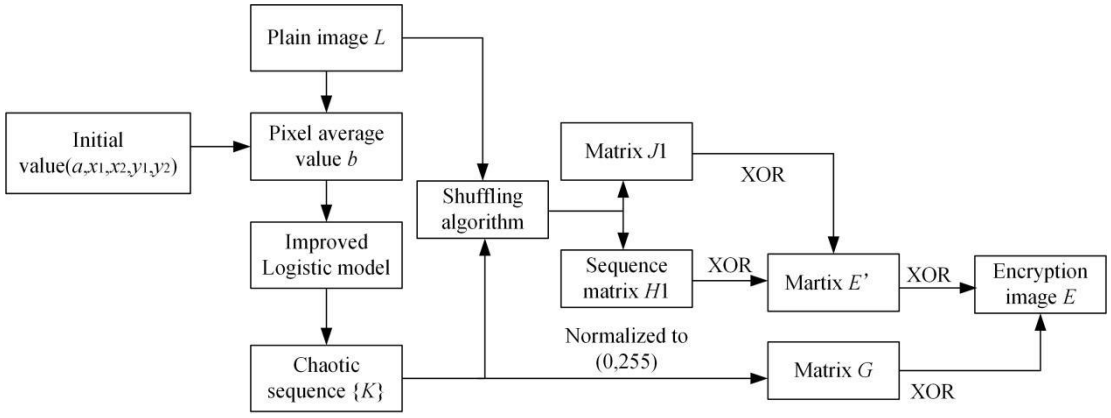

Figure 9. Flowchart of the encryption process.

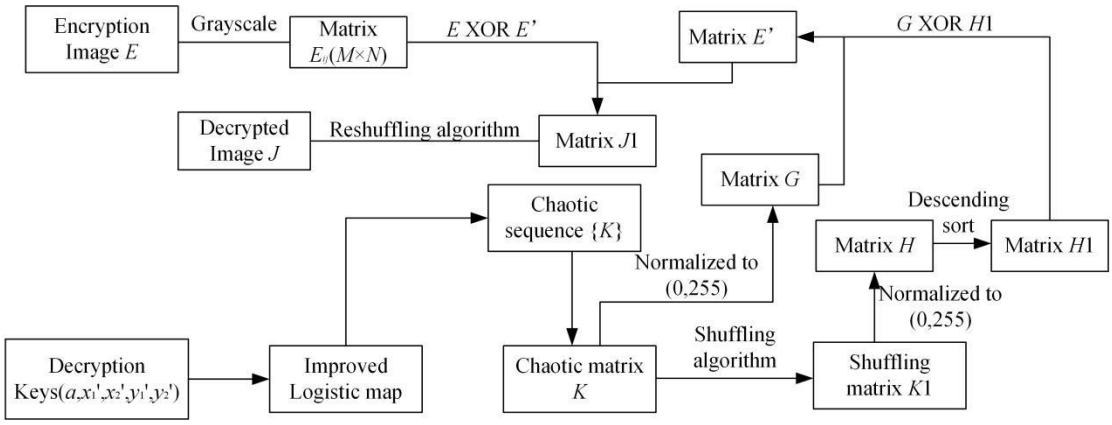

Figure 10. Flowchart of the decryption process.

4.2. Security Performances Tests

In this section, various analyses are performed to prove the effectiveness of the encryption algorithm. In these tests, we set a = 3.999, x_1 = 0.58432, x_2 = 0.54752, y_1 = 0.35421, and y_2 = 964,326.

4.2.1. Encryption and Decryption Experiments

In this section, we take the grayscale images 'Horse' and 'Baboon' as examples and perform operations including encryption, correct decryption, and incorrect decryption, the results of which are shown in Figure 11. Among them, incorrect decryption consists of changing the initial value slightly, by 10^{-10}, and subsequently verifying the decryption effect. It can be clearly seen from Figure 11 that the encryption algorithm proposed in this paper is able to effectively encrypt and decrypt the image. When the key is wrong, the

decrypted image is still full of noise and cannot be recognized, proving that this scheme is secure.

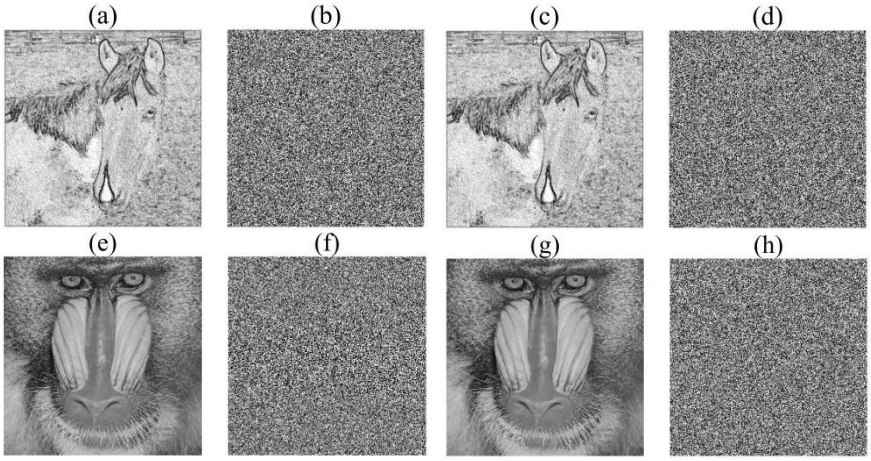

Figure 11. Encryption and decryption: (**a**) 'Horse' image; (**b**) encrypted image; (**c**) image decrypted with correct key; (**d**) image decrypted with incorrect key; (**e**) 'Baboon' image; (**f**) encrypted image; (**g**) image decrypted with correct key; (**h**) image decrypted with incorrect key.

4.2.2. Histogram Analysis

Histograms are an intuitive method for observing the intensity distribution of image pixels, through which the gray distribution can be obtained. The histograms from two images (i.e., the original 'Horse' and the encrypted 'Horse') are shown in Figure 12. As can be seen from Figure 12a, the histogram obtained from the original image possesses an uneven distribution. Meanwhile, it can be seen in Figure 12b that the histogram from the encrypted image possesses an almost consistent distribution. This result proves that the proposed algorithm can cause the pixel distribution of the encrypted image to be uniform, showing that the encrypted image possesses effective resistance against statistical attacks.

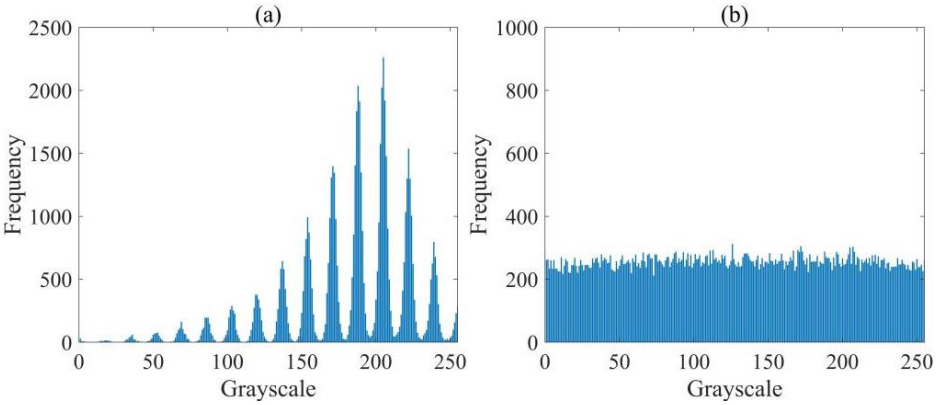

Figure 12. Images: (**a**) original 'Horse' image; (**b**) encrypted image.

4.2.3. Correlation Analysis

Ideal encryption images will have a lower correlation of neighbor pixels, which is reflected in the relevant coefficient. The measurement formula for the correlation coefficient ρ_{xy} is as follows:

$$\rho_{xy} = \frac{\sum_{i=1}^{Q}\left(x_i - \frac{1}{Q}\sum_{i=1}^{Q} x_i\right)\left(y_i - \frac{1}{Q}\sum_{i=1}^{Q} y_i\right)}{\sqrt{\sum_{i=1}^{Q}\left(x_i - \frac{1}{Q}\sum_{i=1}^{Q} x_i\right)^2 \times \sum_{i=1}^{Q}\left(y_i - \frac{1}{Q}\sum_{i=1}^{Q} y_i\right)^2}} \tag{21}$$

where x_i and y_i are the data sequences composed of adjacent pixels, and Q is the length of the sample sequence. Figure 13a–f display the adjacent pixel distribution of the original 'Horse' and the encrypted 'Horse' image in different directions (i.e., horizontal, vertical, and diagonal). From Figure 13a–c, it can be seen that the distribution points are clustered near the diagonal line, indicating that neighboring pixels with strong correlations appear in the original image. According to Figure 13d–f, points are distributed with no special appearance, indicating that there is no correlation between neighboring pixels in the encrypted image. Observing the correlation coefficients in Table 1, the absolute values of our correlation coefficient in three directions are all close to 0, which is also better than the results obtained for some other encryption algorithms. Thus, our encryption algorithm can reduce the pixel correlation of plain images and has strong resistance to correlation attacks.

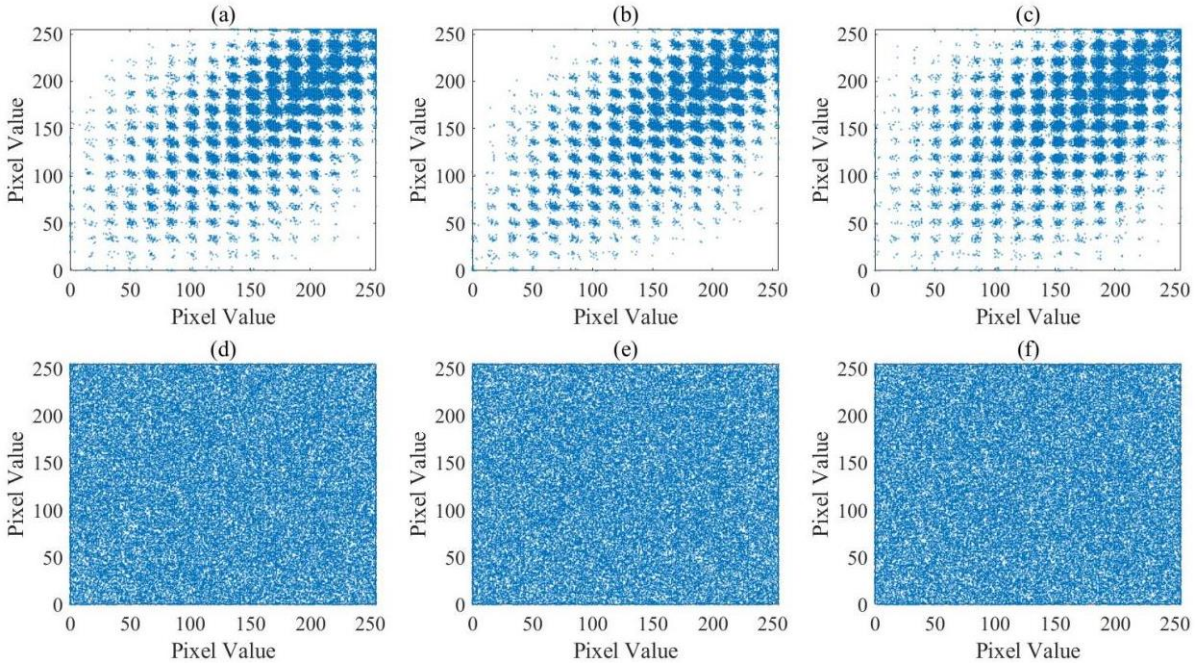

Figure 13. Distribution of adjacent pixels: horizontal direction: (**a**) Horse; (**d**) encrypted; vertical direction: (**b**) Horse; (**e**) encrypted; diagonal direction: (**c**) Horse; (**f**) encrypted.

Table 1. Correlation coefficients.

Images	Horizontal	Vertical	Diagonal
Plain Horse image	0.6425	0.6682	0.5179
Encrypted Horse image	−0.0037	0.0076	0.0016
Xiang, H. et al. [12]	−0.0084	−0.0003	−0.0089
Liu, L.F. et al. [15]	0.0042	−0.0021	−0.0043
Ouannas et al. [19]	−0.0055	0.0045	−0.0055
Zhang, S. et al. [20]	−0.0033	0.0094	0.0021
Wu, J. et al. [21]	0.0056	0.0037	0.0032

4.2.4. Key Sensitivity

Key sensitivity means that the image encryption and decryption will undergo a notable change if the secret key is altered slightly. At the same time, keys are provided that add the average value of the plain-text image, so that the key will have plain-text correlation and should be resistant to known plain-text attacks. Here, we performed a visualization experiment with minor modifications to the initial values (a, x_1, x_2, y_1, and y_2), and the decrypted images are shown in Figure 14b–f. From the resulting figure, it can be seen that when the initial values and parameters are changed by 10^{-10} separately, the decrypted

images cannot be restored to the initial image. Furthermore, the following equation was used to calculate the mean square error (MSE) to carry out the numerical experiments.

$$MSE = \frac{\sum_i^M \sum_i^N (w_i - o_i)^2}{M \times N} \tag{22}$$

where $M \times N$ is the size of the image, o_i is the modified image pixel, and w_i is the original pixel. Figures 15 and 16 display a comparison of the MSE for displaying encrypted images and decrypted images when using our original key and the changed key, respectively. As can be seen from Figures 15 and 16, the MSE changes significantly when the secret key deviates slightly, which indicates that the scheme in this paper has outstanding key sensitivity for both encryption and decryption algorithms.

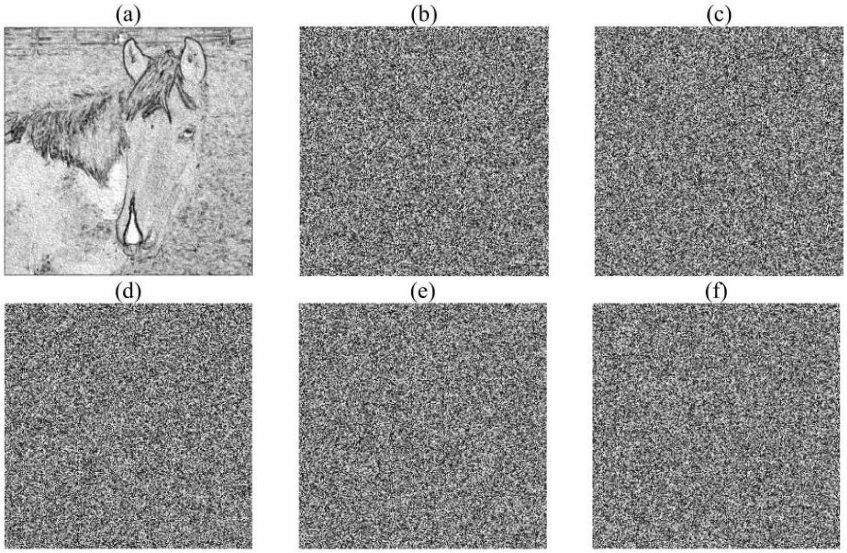

Figure 14. (**a**) Original image; (**b**) decrypted image with $a - 10^{-10}$; (**c**) decrypted image with $x_1 - 10^{-10}$; (**d**) decrypted image with $x_2 - 10^{-10}$; (**e**) decrypted image with $y_1 - 10^{-10}$; (**f**) decrypted image with $y_2 - 10^{-10}$.

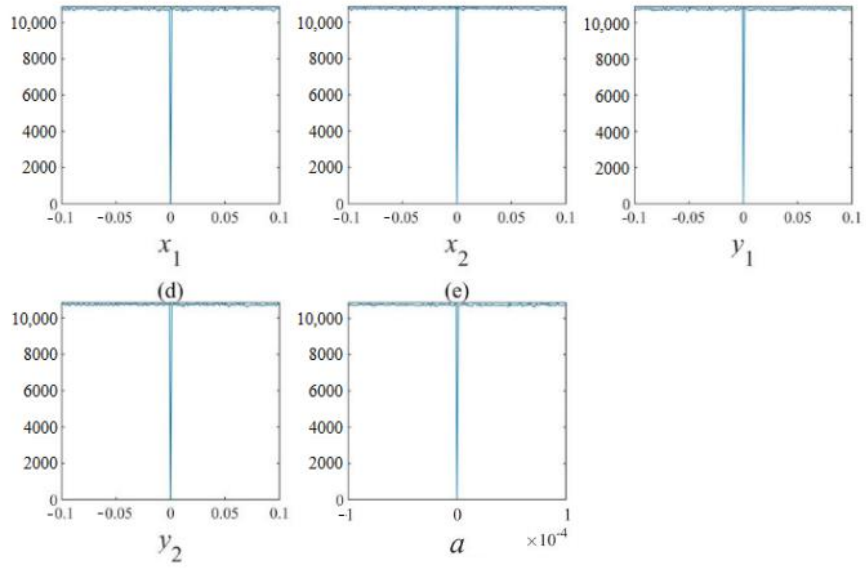

Figure 15. MSE analysis of encryption process: (**a**) x_1; (**b**) x_2; (**c**) y_1; (**d**) y_2; (**e**) a.

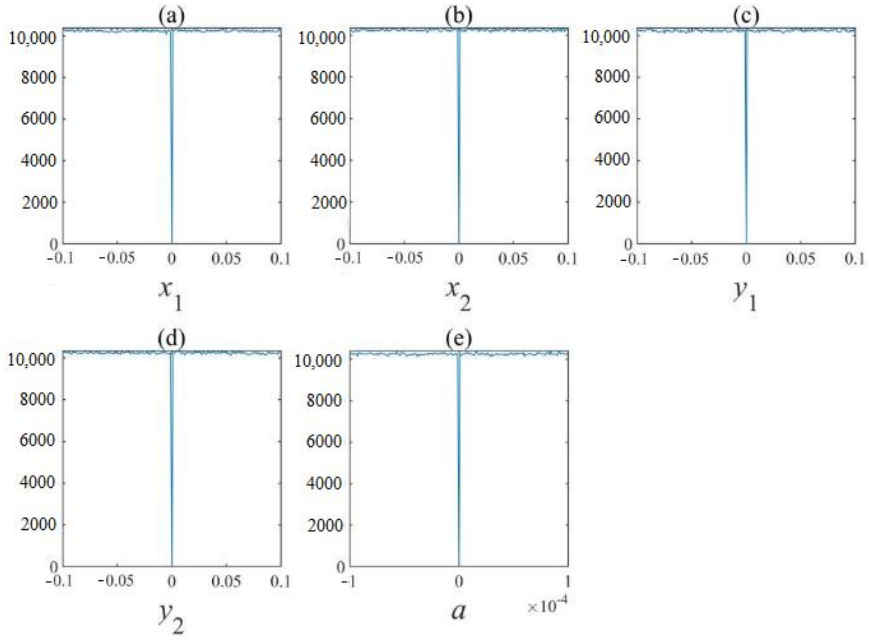

Figure 16. MSE analysis of decryption process: (**a**) x_1; (**b**) x_2; (**c**) y_1; (**d**) y_2; (**e**) a.

4.2.5. Key Space Analysis

In an ideal cryptographic system, the key space should be large enough (no less than 2^{128}) to resist brute force attacks. In this image encryption algorithm, the initial values x_1, x_2, y_1, y_2, and the control parameter a are used as the secret keys. Assuming that the highest computer accuracy is 10^{-14}, the key space can be approximately estimated as $10^{14} \times 10^{14} \times 10^{14} \times 10^{14} \times 10^{14} = 10^{70} \approx 2^{233}$. Table 2 compares the key spaces of various encryption algorithms. From Table 2, it can be seen that the key spaces are all greater than 2^{128}, which demonstrates that the existing algorithms have sufficient key space to resist brute force attacks. Furthermore, the key space in our encryption algorithm is greater than the others, indicating that our algorithm is more competitive in this regard.

Table 2. Key space.

Algorithms	Space
Proposed scheme	2^{233}
Tang, J. et al. [18]	2^{231}
Zhang, Y. et al. [22]	2^{160}
Chen, C. et al. [23]	2^{152}
Zhang, Y.Q. et al. [24]	2^{186}

4.2.6. Robustness Analysis

Robustness testing aims to detect whether the proposed method can restore the original image when the image pixels have been partially destroyed. In a secure encryption algorithm, the original image will be able to be recovered even if some pixel data are lost or filled with some noise. In this test, we added varying degrees of 'salt and pepper' noise to the encrypted "Horse" image and performed a data loss attack, as shown in Figure 17a_1–d_1. As can be seen from Figure 17a_2–d_2, when using our decryption algorithm, the four ciphertext images subjected to different attacks could all be restored to a state in which their original images are recognizable. These results prove that the encryption scheme proposed in this paper is robust against noise and data loss attacks.

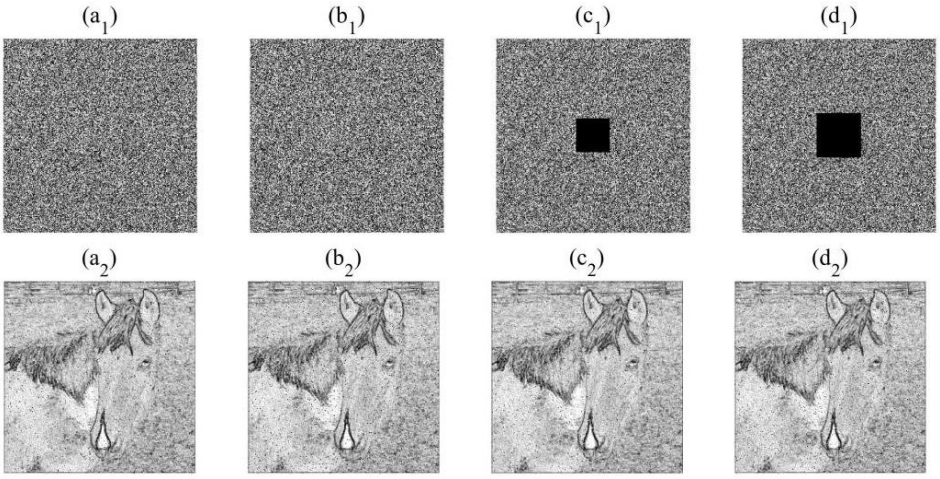

Figure 17. Robustness analysis: (**a1**) 3% 'salt and pepper' noise (encrypted); (**a2**) 3% 'salt and pepper' noise (decrypted); (**b1**) 5% 'salt and pepper' noise (encrypted); (**b2**) 5% 'salt and pepper' noise (decrypted); (**c1**) 3% data loss (encrypted); (**c2**) 3% data loss (decrypted); (**d1**) 5% data loss (encrypted); (**d2**) 5% data loss (decrypted).

4.2.7. Anti-Differential Attack Analysis

The differential attack is a frequent type of attack in cryptographic algorithms. Thus, the ability to resist differential attack must be considered in encryption algorithms. Both NPCR and UACI can be utilized to testify to the ability of the encryption algorithm to stand up to some attacks. For the same gray image, one of the grayscale values is changed in order to obtain two encrypted images. By calculating the percentage and change degree of pixels with different gray values at the same position in the total number of original pixels in two encrypted images, the pixel change rate NPCR and the unified average changing intensity UACI are obtained. For an algorithm with an ideal encryption effect, the ideal value of NPCR is about 0.9961, and the UACI value is near 0.3346. The NPCR and UACI can be calculated as follows.

$$NPCR = \frac{\sum_i^M \sum_j^N H(i,j)}{M \times N} \times 100\% \tag{23}$$

$$UACI = \frac{1}{M \times N}\left(\sum_i^M \sum_j^N \frac{I(i,j) - I'(i,j)}{255}\right) \times 100\% \tag{24}$$

where I and I' are two encrypted images with a size $M \times N$.

We randomly modified the pixel of the plain-text image by 1 bit to calculate the values of NPCR and UACI, and the results are shown in Table 3. Table 3 displays the calculated UACI values and NPCR values for the different encryption algorithms. From this table, it can be seen that our encryption algorithm yields UACI values and NPCR values that are extremely close to the desired values, which means that this algorithm is able to effectively resist differential attack, and is competitive with other algorithms in this respect.

Table 3. NPCR and UACI values.

Images	NPCR	UACI
Horse image	0.9959	0.3332
Liu, L.et al. [15]	0.9969	0.3351
Tang, J. et al. [18]	0.9956	0.3328
Zhang. S. et al. [20]	0.9960	0.3265
Wu, J. et al. [21]	0.9962	0.3341

4.2.8. Information Entropy Analysis

By calculating the information entropy, the pixel randomness of the encrypted image can be evaluated. Information entropy $G(x)$ can be described as follows:

$$G(x) = \sum_i^n m(x_i)\log_2 \frac{1}{m(x_i)} \quad (25)$$

where n is the total number of symbols, and $m(x_i)$ is the probability of symbol x_i. Generally, the $G(x)$ of an ideal encrypted image will approach a value of 8. The $G(x)$ values of our encryption algorithm and some other methods are shown in Table 4. From Table 4, it can be seen that the information entropy of the encrypted Horse image is 7.9970, which is close to the ideal value of 8, and is not much different from the other algorithms. This result implies that our encryption algorithm can effectively make the original image random-like.

Table 4. Information entropy in various algorithm.

Images	Information Entropy
Plain Horse image	6.5645
Encrypted Horse image	7.9970
Xiang, H. et al. [12]	7.9963
Tang, J. et al. [18]	7.9971
Wu, J. et al. [21]	7.9976
Yu, J. et al. [25]	7.9973
Zhang, X. et al. [26]	7.7841

4.2.9. Speed Analysis

In an effective encryption scheme, the encryption speed, which is related to the real-time applicability of the algorithm, needs to be considered. The test environment used in this paper was MATLAB 2019, installed on a computer with a 2.80 GHz CPU and 16 GB RAM. The execution speed of the algorithm proposed in this paper for encryption operations on grayscale images of 256 × 256 sizes is shown in Table 5. As can be seen from Table 5, our image encryption algorithm has a faster execution speed than the other schemes. This also shows that the scheme proposed in this paper has superior performance in terms of both security and practicality.

Table 5. Speed tests.

Algorithms	Images	Time (/s)	Speed (Mb/s)
Proposed	Horse	0.0225	22.222
Proposed	Baboon	0.0236	21.186
Wang, J. et al. [27]	Horse	0.6033	0.8287
Li, B. et al. [28]	Lena	0.3263	1.5323
Mondal, B. et al. [29]	Baboon	1.5069	1.3272
Yousif, S.F. et al. [30]	Baboon	1.634973	1.2232

5. Conclusions

In this paper, a method of introducing delayed state and coupling was proposed with the aim of improving the chaotic characteristics of chaotic maps. By using the structure of a chaotic function perturbed by a delayed state, two chaotic maps are coupled and caused to interact with each other. The results of numerical simulations demonstrate that the complexity and security of the chaotic system can be improved effectively using the delay coupling method. Furthermore, in order to demonstrate the practicability of this method, a novel image encryption algorithm was proposed based on the delayed coupled Logistic map. Chaotic sequences with high pseudo-randomness were used to perform the shuffling-diffusing operation on a plain-text image, and finally the image was effectively encrypted

and protected. Meanwhile, the algorithm proposed in this paper can be implemented using electronic devices, and has excellent applicability.

Author Contributions: Conceptualization, L.L.; Methodology, S.H.; Software, S.H.; Validation, G.D. and X.L.; Formal analysis, G.D. and X.L.; Investigation, L.L.; Resources, L.L. and X.L.; Data curation, L.L.; Writing—original draft, S.H.; Writing—review & editing, G.D. and L.L.; Visualization, X.L.; Supervision, G.D.; Project administration, X.L.; Funding acquisition, L.L. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (6236070219, 62262039, 62262023), the Outstanding Youth Foundation of Jiangxi Province (20212ACB212006), the Finance Science and Technology Special "Contract System" Project of Jiangxi Province (ZBG20230418014), the Science and Technology Innovation Platform Project of Jiangxi Province (20181BCD40005), the Jiangxi Province Natural Science Foundation of China (20192BAB207019), the Practice Innovation Training Program of Jiangxi Province for College Students (202310403276, 202310403277, S202310403274, S202310403275, S202310403282), the Science and Technology Research Support Project of Jiangxi Provincial Education Department (GJJ2210701), and the Jiangxi Province Educational Reform Key Project (JXJG-2020-1-2).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Maritz, M.F. A note on exact solutions of the logistic map. *Chaos* **2020**, *30*, 033136. [CrossRef]
2. Anees, A.; Hussain, I.; Alkhaldi, A.H.; Aslam, M.; Siddiqui, N.; Ahmed, R. Image encryption based on Chebyshev chaotic map and S8 S-boxes. *Opt. Appl.* **2019**, *2019*, 317–330.
3. Liu, J.; Zhong, M.; Liu, B.; Liu, Y.; Li, B. Design of three-dimensional dynamic integer tent map and its image encryption algorithm. *Multimed. Tools Appl.* **2021**, *80*, 19219–19236.
4. Zou, C.; Zhang, Q.; Wei, X.; Liu, C. Image Encryption Based on Improved Lorenz System. *IEEE Access* **2020**, *8*, 75728–75740. [CrossRef]
5. Zhao, H.; Xie, S.; Zhang, J.; Wu, T. Efficient image encryption using two-dimensional enhanced hyperchaotic Henon map. *J. Electron. Imaging* **2020**, *29*, 023007. [CrossRef]
6. Wang, X.; Lin, S.; Li, Y. A chaotic image encryption scheme based on cat map and MMT permutation. *Mod. Phys. Lett. B* **2019**, *33*, 1950326. [CrossRef]
7. Jia, H.; Wang, Q. Image Encryption Implementation Based on Fractional-order Chen System. In Proceedings of the International Conference on Artificial Life and Robotics, Okinawa, Japan, 29–31 January 2016; Volume 21, pp. 254–257.
8. Yang, Y.G.; Guan, B.W.; Zhou, Y.H.; Shi, W.M. Double image compression-encryption algorithm based on fractional order hyper chaotic system and DNA approach. *Multimed. Tools Appl.* **2020**, *80*, 691–710. [CrossRef]
9. Barboza, R. Dynamics of a hyperchaotic Lorenz system. *Int. J. Bifurc. Chaos* **2007**, *17*, 4285–4294. [CrossRef]
10. Yang, Y.G.; Zou, L.; Zhou, Y.H.; Shi, W.M. Visually meaningful encryption for color images by using Qi hyper-chaotic system and singular value decomposition in YCbCr color space. *Optik* **2020**, *213*, 164422. [CrossRef]
11. Tang, J.; Zhang, Z.; Chen, P.; Zhang, F.; Ni, H.; Huang, Z. An image layered scrambling encryption algorithm based on a novel discrete chaotic map. *IET Image Process.* **2023**, *17*, 518–532. [CrossRef]
12. Xiang, H.; Liu, L. An improved digital logistic map and its application in image encryption. *Multimed. Tools Appl.* **2020**, *79*, 30329–30355. [CrossRef]
13. Liu, L.; Xiang, H.; Li, X. A novel perturbation method to reduce the dynamical degradation of digital chaotic maps. *Nonlinear Dyn.* **2021**, *103*, 1099–1115. [CrossRef]
14. Liu, B.; Xiang, H.; Liu, L. Reducing the Dynamical Degradation of Digital Chaotic Maps with Time-Delay Linear Feedback and Parameter Perturbation. *Math. Probl. Eng.* **2020**, *2*, 4926937. [CrossRef]
15. Liu, L.; Miao, S. Delay-introducing method to improve the dynamical degradation of a digital chaotic map. *Inf. Sci.* **2017**, *396*, 1–13. [CrossRef]
16. Liu, L.; Liu, B.; Hu, H.; Miao, S. Reducing the Dynamical Degradation by Bi-Coupling Digital Chaotic Maps. *Int. J. Bifurc. Chaos* **2018**, *28*, 1850059. [CrossRef]
17. Li, S.; Ding, W.; Yin, B.; Zhang, T.; Ma, Y. A Novel Delay Linear Coupling Logistics Map Model for Color Image Encryption. *Entropy* **2018**, *20*, 463. [CrossRef]
18. Tang, J.; Yu, Z.; Liu, L. A delay coupling method to reduce the dynamical degradation of digital chaotic maps and its application for image encryption. *Multimed. Tools Appl.* **2019**, *78*, 24765–24788. [CrossRef]

19. Ouannas, A.; Khennaoui, A.-A.; Bendoukha, S.; Wang, Z.; Pham, V.-T. The Dynamics and Control of the Fractional Forms of Some Rational Chaotic Maps. *J. Syst. Sci. Complex.* **2020**, *33*, 26–45. [CrossRef]
20. Zhang, S.; Gao, T. An image encryption scheme based on DNA coding and permutation of hyper-image. *Multimed. Tools Appl.* **2016**, *75*, 17157–17170. [CrossRef]
21. Wu, J.; Liao, X.; Bo, Y. Image encryption using 2D Hénon-Sine map and DNA approach. *Signal Process.* **2018**, *153*, 11–23. [CrossRef]
22. Zhang, Y.; Xu, B.; Zhou, N. A novel image compression-encryption hybrid algorithm based on the analysis sparse representation. *Opt. Commun.* **2017**, *392*, 223–233. [CrossRef]
23. Chen, C.; Sun, K.; Peng, Y.; Alamodi, A.O.A. A novel control method to counteract the dynamical degradation of a digital chaotic sequence. *Eur. Phys. J. Plus* **2019**, *134*, 31. [CrossRef]
24. Zhang, Y.Q.; Huang, H.F.; Wang, X.Y.; Huang, X.H. A secure image encryption scheme based on genetic mutation and MLNCML chaotic system. *Multimed. Tools Appl.* **2021**, *80*, 19291–19305. [CrossRef]
25. Yu, J.; Li, C.; Song, X.; Guo, S.; Wang, E. Parallel Mixed Image Encryption and Extraction Algorithm Based on Compressed Sensing. *Entropy* **2021**, *23*, 278. [CrossRef]
26. Zhang, X.; Wang, X. Multiple-image encryption algorithm based on mixed image element and permutation. *Comput. Electr. Eng.* **2017**, *62*, 6–16. [CrossRef]
27. Wang, J.; Liu, L. A novel chaos-based image encryption using magic square scrambling and octree diffusing. *Mathematics* **2022**, *10*, 457. [CrossRef]
28. Li, B.; Liu, J.; Liu, Y.; Xu, H.; Wang, J. Image encryption algorithm with 2D coupled discrete chaos. *Multimed. Tools Appl.* **2023**, 1–22. [CrossRef]
29. Mondal, B.; Singh, J.P. A lightweight image encryption scheme based on chaos and diffusion circuit. *Multimed. Tools Appl.* **2022**, *81*, 34547–34571. [CrossRef]
30. Yousif, S.F.; Abboud, A.J.; Alhumaima, R.S. A new image encryption based on bit replacing, chaos and DNA coding techniques. *Multimed. Tools Appl.* **2022**, *81*, 27453–27493. [CrossRef]