



Article

Plaintext-Related Dynamic Key Chaotic Image Encryption Algorithm

Zeming Wu , Ping Pan, Chunyang Sun  and Bing Zhao *

Electronic Engineering College, Heilongjiang University, Harbin 150080, China; 2201669@s.hlju.edu.cn (Z.W.); 2201647@s.hlju.edu.cn (P.P.); 2191309@s.hlju.edu.cn (C.S.)

* Correspondence: zhaobing@hlju.edu.cn

Abstract: To address the problems of the high complexity and low security of the existing image encryption algorithms, this paper proposes a dynamic key chaotic image encryption algorithm with low complexity and high security associated with plaintext. Firstly, the RGB components of the color image are read, and the RGB components are normalized to obtain the key that is closely related to the plaintext, and then the Arnold transform is used to stretch and fold the RGB components of the color image to change the position of the pixel points in space, so as to destroy the correlation between the adjacent pixel points of the image. Next, the generated sequences are independently encrypted with the Arnold-transformed RGB matrix. Finally, the three encrypted images are combined to obtain the final encrypted image. Since the key acquisition of this encryption algorithm is related to the plaintext, it is possible to achieve one key per image, so the key acquisition is dynamic. This encryption algorithm introduces chaotic mapping, so that the key space size is 10^{180} . The key acquisition is closely related to the plaintext, which makes the ciphertext more random and resistant to differential attacks, and ensures that the ciphertext is more secure after encryption. The experiments show that the algorithm can encrypt the image effectively and can resist attack on the encrypted image.

Keywords: image encryption; plaintext-related; dynamic keys; chaotic systems



Citation: Wu, Z.; Pan, P.; Sun, C.; Zhao, B. Plaintext-Related Dynamic Key Chaotic Image Encryption Algorithm. *Entropy* **2021**, *23*, 1159. <https://doi.org/10.3390/e23091159>

Academic Editors: Bellie Sivakumar and José A. Tenreiro Machado

Received: 2 August 2021

Accepted: 30 August 2021

Published: 2 September 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of internet technology, a large amount of image information is transmitted in the network, and people can gain the required image information quickly through the internet. However, the internet is an open platform, and images can be easily filched or attacked during their transmission, so finding how to transmit this information securely has become an urgent issue, and also makes digital image encryption one of the research hotspots in the field of encryption. Digital image encryption is an important means to protect image information, and it is of great significance to protect the transmission security of digital images. In recent years, many researchers have mainly studied gray images and color images. Compared with gray images, color images contain rich information, and the requirements for the security of encryption algorithms are more urgent. Due to the non-periodic behavior of chaos, the sensitivity to initial values, and the unpredictability, the chaos theory has been widely used in the image encryption process. Robert Matthews first proposed the idea of encryption based on the chaos theory [1]. Since then, chaos has been widely used in image encryption [2–5]. So far, many research results exist that are based on chaotic image encryption, such as the Hopfield chaotic neural network [6–8], which is a typical dynamic neural network with rich dynamic properties; however, the self-feedback Hopfield network used to generate chaotic phenomena is complex in its structure, computationally intensive with fixed parameters, DNA encryption [9–12], DNA computation with huge parallelism, and has huge storage and ultra-low power consumption. The compressed sensing (CS) [13–16] compression feature allows multimedia encryption

schemes with a much reduced length of ciphertext, and simple linear measurements make the encryption process very efficient. Therefore, CS-based image encryption schemes have also attracted a lot of attention in recent years. For wavelet transform (WT) [17,18], in digital image encryption schemes, wavelet packet transform (WPT) is often used in the preprocessing stage to decompose the plaintext image, to obtain different image signal components, in order to eliminate some negative factors that are detrimental to the subsequent processing steps and to improve the overall operational efficiency of the image processing scheme. S-box systems [19,20] are used to replace the substitution cipher structure, as a way to ensure the obfuscation performance of the block cipher and the nonlinearity of the generated image, thus improving the performance of the encrypted image. Hyper-chaotic systems [21,22], with multiple initial values and parameters, can greatly improve the key space of encryption algorithms, but the corresponding algorithm complexity increases. Hash functions [23–25] play an important role in image encryption systems, and, due to the irreversibility of hash functions, they can resist known plaintext attacks, selective plaintext attacks, and selective ciphertext attacks. In terms of quantum color image encryption schemes [26–29], quantum-based encrypted images will play an important role as specific and critical quantum information types in the future era of quantum computers. As well as other methods, the literature [30] proposed an encryption, compression and transmission scheme. The scheme is based on a fractional-order chaotic system combined with discrete wavelet transform (DWT) and quadrature phase-shift keying (QPSK) modulation. The cipher performs multiple rounds of digital operations between the vector state of the fractional-order chaotic system and the original image. The transmission process is implemented between a pair of software-defined radio modules, through a QPSK modulation scheme. The literature [31] proposes a two-color image encryption algorithm based on two-dimensional compression perception and wavelet basis, which can simultaneously implement image encryption and compression, and this scheme can effectively reduce the amount of data, and improve the efficiency of transmitting data and distributing keys. However, the complexity is high. The literature [32] proposed a hybrid chaotic fractal system based on a Julia fractal set, and a 3D Lorentzian chaotic system composed of a hybrid chaotic fractal system that generates encrypted images by confusing and diffusing the original image, which has lower computational complexity and provides higher security. The literature [33] proposed a hybrid technique combining Mersenne Twister (MT), deoxyribonucleic acid (DNA), and chaotic dynamical Rossler system (MT-DNA-Chaos), in which the three encryption algorithms not only improve the overall efficiency of data randomization, but the algorithm is more flexible in computation and more secure. The literature [34] proposed a new discrete chaotic encryption algorithm based on DNA coding and SHA-256, but the structure is more complex and there is still room to improve the information entropy after encryption. The literature [35] proposed a novel encryption algorithm based on multiple chaotic mappings, to encrypt the regions of interest (ROI) in images, and this method greatly improves the speed and performance of encryption. The literature [36] introduced an improved Hénon mapping with richer chaotic behavior and better complexity, to improve the security of the color image encryption algorithm, but it uses two-dimensional chaotic mapping with a small key space. Some classical attacks on recently proposed encryption techniques (retrieving the key in a few computations by using a choice plaintext attack and a known plaintext–ciphertext pair) were conducted in the literature [37], to discover the flaws in some of these design structures and suggest some improvements.

Combining the above literature and addressing some of the problems that appear in the literature, this paper proposes a simple color image encryption algorithm that is closely related to the plaintext image and has high security with low complexity. The algorithm introduces the plaintext information and chaotic mapping, which not only improves the key space greatly, but, also, the encrypted result is closely related to the plaintext image, and the key of each image can be guaranteed to be completely different. Moreover, it improves the performance of encrypted image randomness and resistance to differential

attack, which ensures that the encrypted ciphertext is more secure. Because the acquisition of the key is closely related to the plaintext, the encryption algorithm proposed in this paper can resist the selected plaintext attack and the known plaintext attack. When the plaintext changes after the dislocation and diffusion operation of the encryption algorithm, the obtained key will also change dynamically, and the correct key will not be obtained. The experiments show that the encryption algorithm proposed in this paper can effectively resist a variety of attacks on encrypted images.

2. Basic Theory

This section describes the two chaotic systems, Lorenz system [38] and Arnold mapping [39], required for the encryption algorithm proposed in this paper.

2.1. Lorenz System

The Lorenz system, discovered by the American scientist Lorenz in 1963 while studying weather forecasting, is the world's first dynamical system that exhibits singular attractors and possesses an extremely rich and complex nonlinear dynamical behavior. The dynamic equation of the three-dimensional Lorenz system is shown in Equation (1), as follows:

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = cx - xz - y \\ \frac{dz}{dt} = xy - bz \end{cases} \quad (1)$$

where a , b , and c are system parameters, taking any value greater than zero, often taking $a = 10$, $b = 8/3$, and c as variables. When $c > 24.74$, the system enters a chaotic state. When $c = 28$, the system enters the optimal chaotic state [40]. In this state, when the initial value is $x(1) = 0.1$, $y(1) = 1$ and $z(1) = 1$, the chaotic attractors generated are shown in Figure 1.

The variation in the three variables x , y , and z with time is nonperiodic and unpredictable, and, as can be observed from the figure, the dynamical orbit is a double helix structure in three dimensions. The helix curves are always confined to a finite space on a plane and they never cut.

From Figure 1, we can see that the values of $x(t)$, $y(t)$, and $z(t)$ take a good randomness after a certain number of iterations. Therefore, when performing data interception, it is better to avoid the more preceding data, to eliminate the initial state effect. In this paper, the interception starts from the 1000th point to ensure the randomness of the data.

The chaotic sequences generated by this system have the following advantages: first, the dynamics of the system are more complex than those of the low-dimensional chaotic system, and the resulting numerical sequences have better randomness; second, there are six parameters and initial values of the system, all of which can be used as keys, which greatly enhances the key space; third, there are three chaotic real-valued sequences, which can meet the demand for the direct chaos displacement of the three parameters proposed in this chapter. It can be observed that the set of dynamical equations of the three-dimensional Lorenz system is a ternary system of ordinary differential equations, which requires the solution of this system of ordinary differential equations, and when it is used in digital image encryption, its numerical solution is required, which comes down to the problem of the numerical solution of ordinary differential equations.

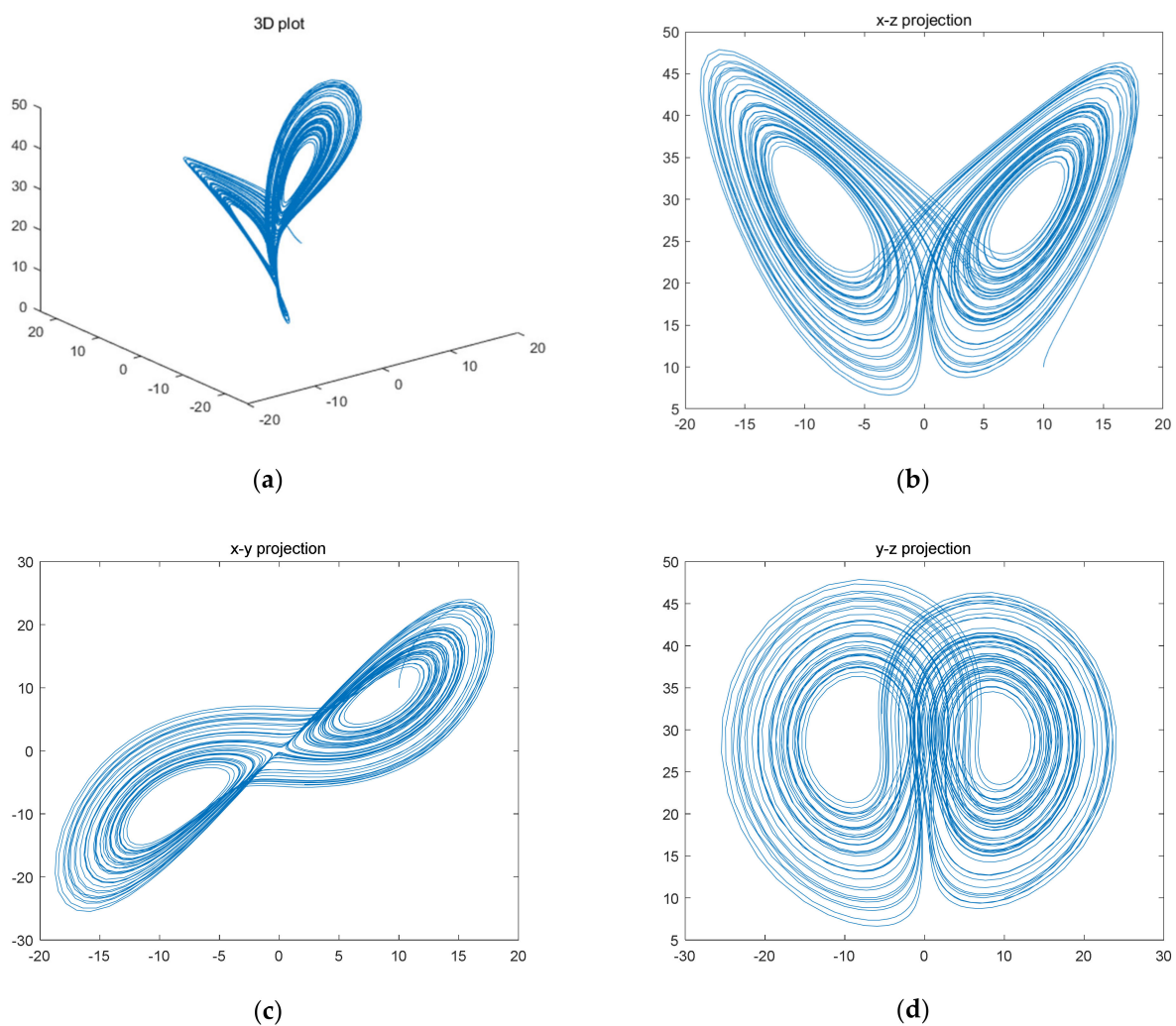


Figure 1. Attractors of Lorenz chaotic system in various dimensions. (a) 3D plot; (b) x-z; (c) x-y; (d) y-z.

2.2. Arnold Mapping

Arnold mapping, also known as Cat mapping, was proposed by the Russian mathematician Vladimir Igorevich Arnold. It is a chaotic mapping method with repeated folding and stretching transformations in a finite area, and is generally used in multimedia chaotic encryption [41].

According to Arnold mapping, the pixels of the original image being transformed are random. However, if the number of iterations is large enough, the original image can eventually be reproduced. This number of iterations is called the Arnold period. The period depends on the size of the image, that is, the Arnold period varies with the size of the image.

Arnold is also considered to be one of the major dislocation algorithms and the algorithm is generated by the following transformation of Equation (2):

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & e \\ d & de + 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod(N). \quad (2)$$

Among them, x_n, y_n denotes the position of the pixel in the grayscale map before the transformation, and x_{n+1}, y_{n+1} denotes the position of the pixel after the transformation, and d, e are positive integer parameters, n denotes the number of current transformations, N is the length or width of the image, and $\bmod(\cdot)$ is the modulus operation.

With the forward transformation formula, this algorithm also needs the inverse transformation formula. The inverse transformation formula is shown in Equation (3) below.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} de + 1 & -e \\ -d & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod(N), \quad (3)$$

The relationship between two matrices transformed into inverse.

The digital image can be seen as a two-dimensional matrix. After the Arnold transformation, the pixel positions of the image will be rearranged, so that the image will be disorganized, thus realizing the scrambling and encryption effect of the image. Arnold transformation has periodicity, that is, after many iterations of Arnold transformation, it will return to the original state. Arnold mapping is generally used for image scrambling. It is mapping from the regular position to a random position, that is, the original strong correlation between the adjacent pixels is replaced by the pixel position, so that the pixels are evenly distributed on the whole image and the correlation between the adjacent pixels is weak. The number of iterations required to repeat the original image is different depending on the size of the image. For a given positive integer N , the period of the transformation is denoted as T_m , and when $N > 2$, the period T_m satisfies $T_m \leq N^2 / 2$. The periods of Arnold transformations for different orders N T_m are shown in Table 1.

Table 1. Periods of Arnold transformations of different orders N T_m [42].

N	2	3	4	5	6	7	8
T_m	3	4	3	10	12	8	6
N	9	10	11	12	25	50	60
T_m	12	30	5	12	50	150	60
N	100	120	125	128	256	480	512
T_m	150	60	250	96	192	240	384

3. Encryption and Decryption Algorithm Design

3.1. Encryption Scheme

The system proposed in this paper mainly has the following two stages: firstly, the pixel position of the original image is changed by Arnold mapping; secondly, the matrix after Arnold transformation is scrambled by using three chaotic sequences generated by the Lorenz chaotic system; finally, the scrambled RGB components are synthesized into the final encrypted image. Figure 2 is the flow chart of the image encryption algorithm designed in this paper. The specific operation process of the encryption algorithm proposed in this paper is summarized as follows:

Step 1: Extract the RGB component of the color image. The extracted RGB is respectively converted into three 8-bit binary matrices, as follows: I_1 , I_2 , and I_3 .

Step 2: Combine the generated matrix I_1 , I_2 , and I_3 , and normalize it to produce the following three keys associated with the plaintext: a_1 , a_2 , and a_3 . These three values are multiplied by the random key q (to enlarge the key space, the value of q can take three different values) as the plaintext-related keys. The multiple keys obtained here are dynamic keys. According to the different images read, the key obtained changes accordingly, that is, the key changes dynamically with the image.

Step 3: set the parameters in the Arnold mapping d , e , N = the width of the image (512 in this paper), the n number of transformations.

Step 4: The generated matrices I_1 , I_2 , and I_3 are transformed by Arnold mapping in step 3, and the RGB matrix components are stretched and folded to change the positions of the pixels in the space, thus destroying the correlation between the adjacent pixel points of the image and generating new matrices I'_1 , I'_2 , and I'_3 .

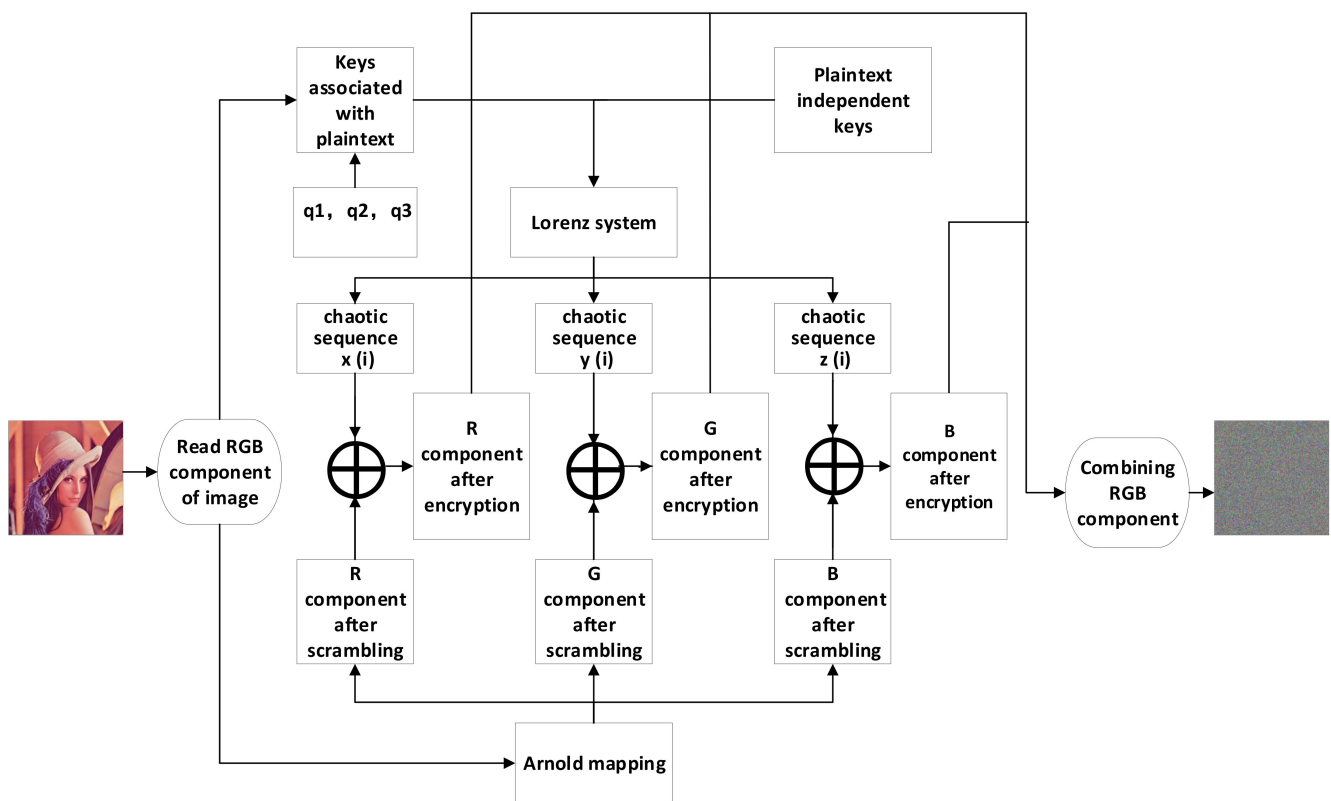


Figure 2. Flow of the designed encryption algorithm.

Step 5: $a'_1, a'_2, a'_3, x_0, y_0, z_0$ are the plaintext-independent keys of the chaotic system. x_0, y_0, z_0 are the initial values of the Lorenz system, respectively, and the parameters of the Lorenz system are generated by Equations (4)–(6). Three chaotic sequences are generated according to the above conditions, and a sequence of length 512×512 is selected as the sequence needed to encrypt the image $x(i), y(i), z(i)$.

$$a'_1 + q_1 a_1 = a, \tag{4}$$

$$a'_2 + q_2 a_2 = b, \tag{5}$$

$$a'_3 + q_3 a_3 = c, \tag{6}$$

Step 6: Convert the resulting chaotic sequence $x(i), y(i), z(i)$ elements in the resulting chaotic sequence into a sequence consisting of data between 0 and 255. The formula used here is shown below:

$$(a(i) \times 1000) \bmod 256, \tag{7}$$

Step 7: the sequences $x(i), y(i)$, and $z(i)$, generated at the end of step 6, are composed from top to bottom and left to right into three 512×512 matrices, which are E_1, E_2 , and E_3 , respectively, as encryption matrices.

Step 8: Convert the E_1, E_2 and E_3 data in the encryption matrix into the same type of data as the color image matrix read in step 1.

Step 9: The matrices E_1, E_2 , and E_3 converted in step 8 are XOR calculated with the matrices I'_1, I'_2 , and I'_3 , generated in step 3, and then the encrypted matrix is combined to generate the final encrypted image.

3.2. Decryption Scheme

The decryption algorithm is the inverse of the encryption algorithm, which first reads the ciphertext image, then uses the key transmitted through the secure channel to obtain the chaotic sequence needed for the decryption process, and then XOR the read ciphertext

data with the chaotic sequence to obtain the final image after the Arnold-transformed key is inverted by the Arnold transform.

4. Simulation Experiments and Performance Analysis

The performance of the proposed image encryption algorithm is analyzed after encrypting and decrypting the Lena color image separately, following the steps given in Section 3. The operating environment of the algorithm is a Windows 10 operating system with a 2.5 GHz Intel CPU I5-4200, 4 GB RAM, and Matlab 2020b.

4.1. Encryption and Decryption

In this section, we use Lena color images (512×512) to evaluate the performance of our scheme. During encryption and decryption, our keys are chosen as $a'_1 = 9.3$, $a'_2 = 27.5$, $a'_3 = 5/3$, $x_0 = 3.21$, $y_0 = 6.27$, and $z_0 = 1.35$ constants $q_1 = q_2 = q_3 = 1$. In Arnold transform, $d = 3$, $e = 9$, $N =$ the width of the image (512 in this paper), $n = 10$. Figure 2 shows the encryption and decryption results using our method. From Figure 3, it can be observed that the Lena picture is successfully encrypted and decrypted, and the encrypted image is completely different with respect to the original image.

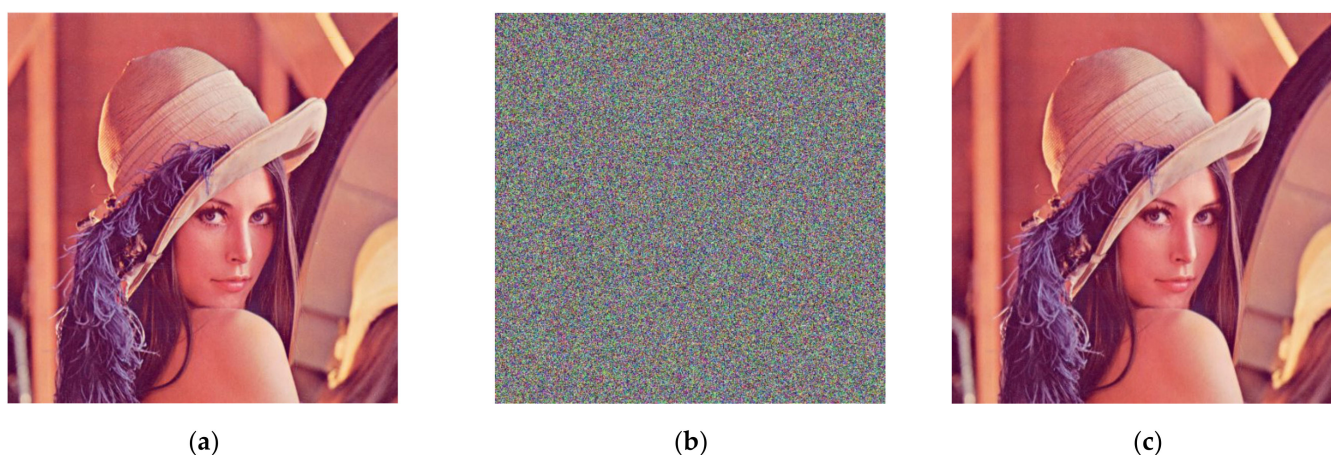


Figure 3. Encrypted and decrypted image of Lena graph. (a) The original image; (b) encryption image; (c) decryption image.

4.2. Histogram Analysis

Histogram analysis is an important security analysis tool in image encryption algorithms, to measure the performance of an encryption scheme in preventing an attacker from accessing the characteristic pixels of an image. Figure 4 shows the histograms of the Lena image (512×512), and Figure 5 shows the histograms of the cryptographic image. It can be observed, from the images, that the encryption scheme produces uniform histograms, which implies that our encryption method has good performance against statistical attacks.

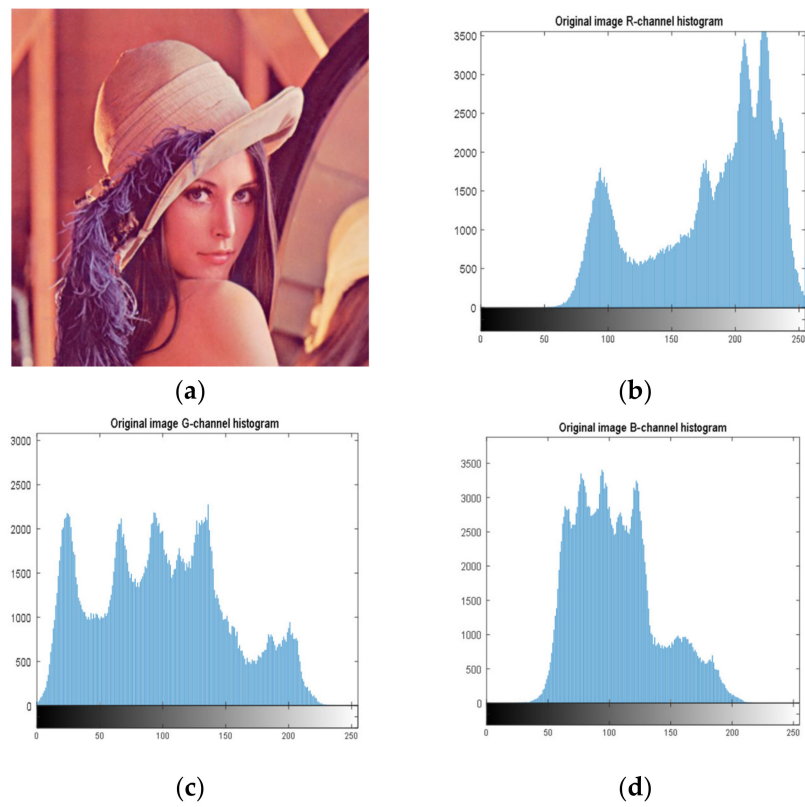


Figure 4. Histogram of components of the original image. (a) Original image; (b) R-channel histogram; (c) G-channel histogram; (d) B-channel histogram.

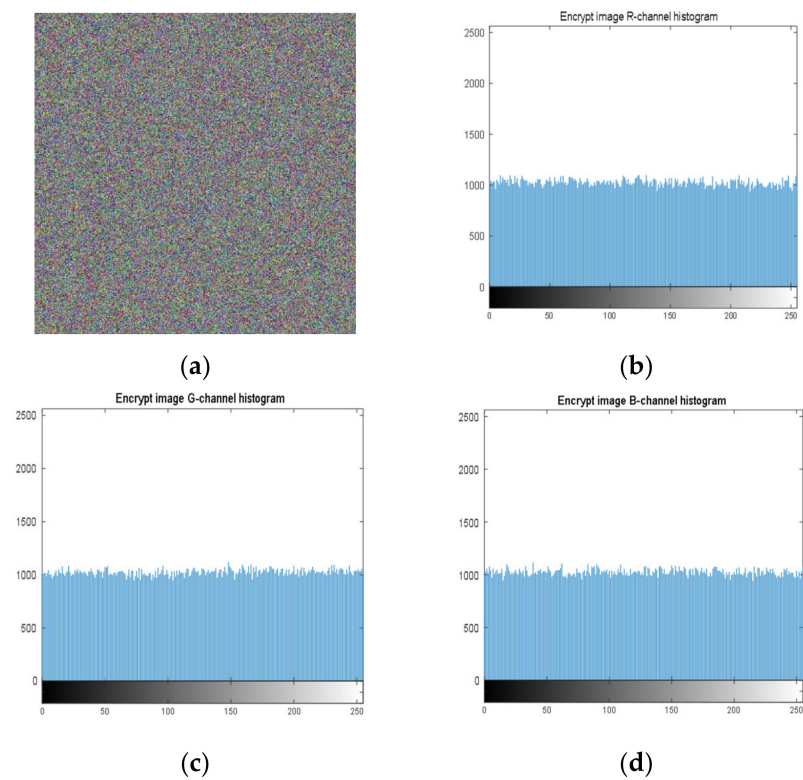


Figure 5. Histogram of each component of an encrypted image. (a) Encrypted image; (b) R-channel histogram; (c) G-channel histogram; (d) B-channel histogram.

4.3. Correlation Analysis

The 8-bit nature of digital images results in high correlation between the adjacent pixels. Image encryption schemes can solve this problem by effective pixel dislocation. In this paper, the image is scrambled by Arnold transform and Lorenz chaotic sequence, and the correlation coefficient is used to measure whether the encryption algorithm proposed in this paper has good performance. A good encryption algorithm should generate an encrypted image with low correlation. In this section, we analyze the images before and after encryption by Equation (8).

$$r(X, Y) = \frac{\text{cov}(X, Y)}{\sigma(X)\sigma(Y)}, \quad (8)$$

where X and Y are two adjacent pixel sequences, $\text{cov}(X, Y)$ is the covariance of X and Y , and $\sigma(\cdot)$ is the standard deviation. When Y is the adjacent horizontal, vertical, and diagonal pixel of each X , $r(X, Y)$ is the horizontal correlation coefficient, vertical correlation coefficient, and diagonal correlation coefficient, respectively. Next, $r(X, Y)$ is used to measure the correlation of two adjacent pixels according to the following criteria: horizontal, vertical, and diagonal. If $r(X, Y)$ is close to one, there is a high correlation between the adjacent pixels. On the contrary, if $r(X, Y)$ is close to zero, the correlation between the adjacent pixels is very small.

Five thousand randomly selected pairs of adjacent points were used to calculate the correlation coefficients for each direction (horizontal, vertical, and diagonal). As observed in Figures 6–8, the pixels of the plain image (512×512) are similar to each other, which depicts that they are highly correlated to each other. However, the adjacent pixels of the encrypted cipher image are not correlated (Figures 9–11). Table 2 shows that the correlation coefficient of the original image is close to one, which indicates that the image has a very high correlation for different directions. However, the correlation coefficients for all directions of the encrypted image are close to zero, which indicates that there is almost no relationship between the neighboring pixels. This implies that our encryption method achieves a good encryption effect. In addition, it can be observed from Table 2 that our method has a lower correlation coefficient than that obtained after encrypting the image in the reference, so it shows that the algorithm proposed in this paper has better resistance to statistical attacks.

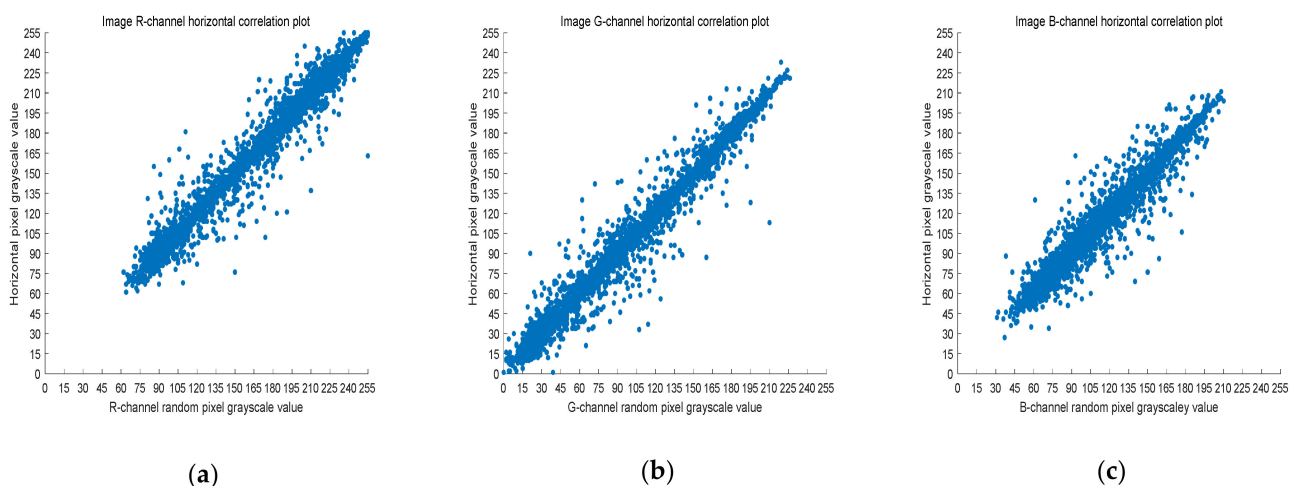


Figure 6. Scatter plot of RGB channels in the horizontal direction of the original image. (a) R-channel scatter plot; (b) G-channel scatter plot; (c) B-channel scatter plot.

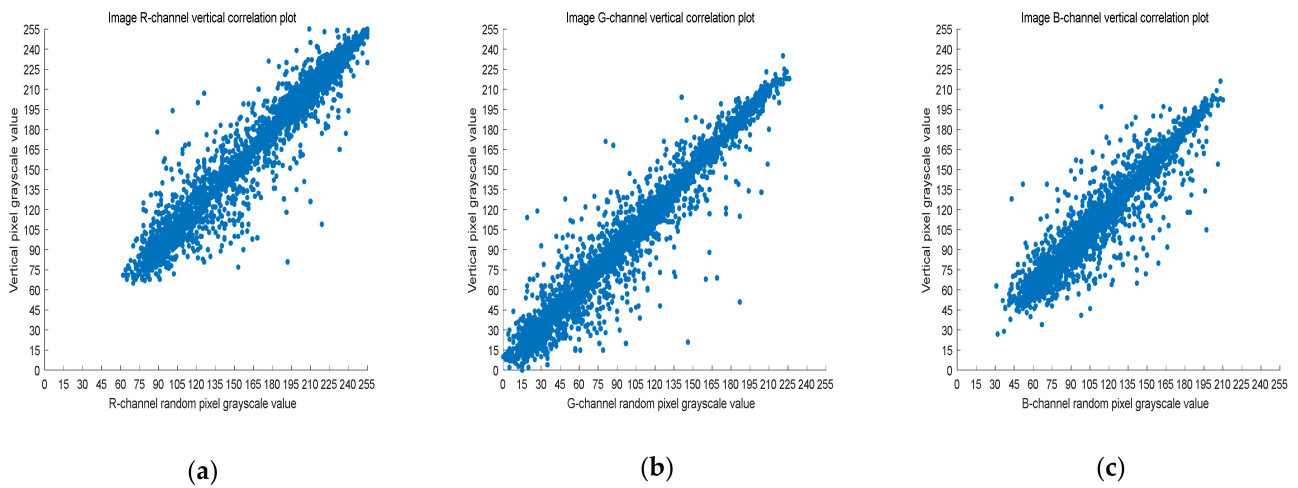


Figure 7. Scatter plot of RGB channels in the vertical direction of the original image. (a) R-channel scatter plot; (b) G-channel scatter plot; (c) B-channel scatter plot.

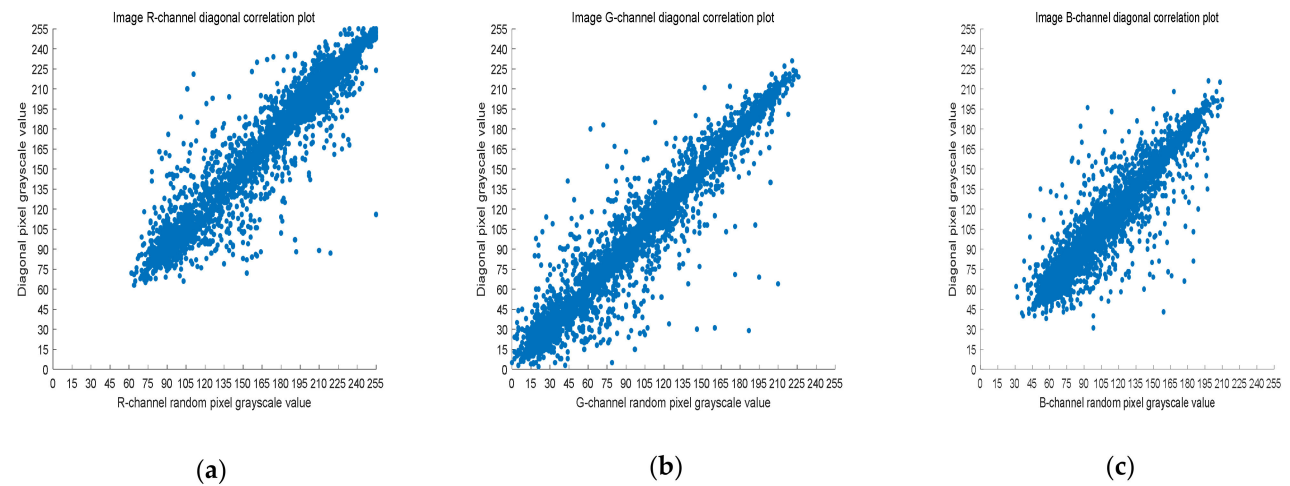


Figure 8. Scatter plot of RGB channels in the diagonal direction of the original image. (a) R-channel scatter plot; (b) G-channel scatter plot; (c) B-channel scatter plot.

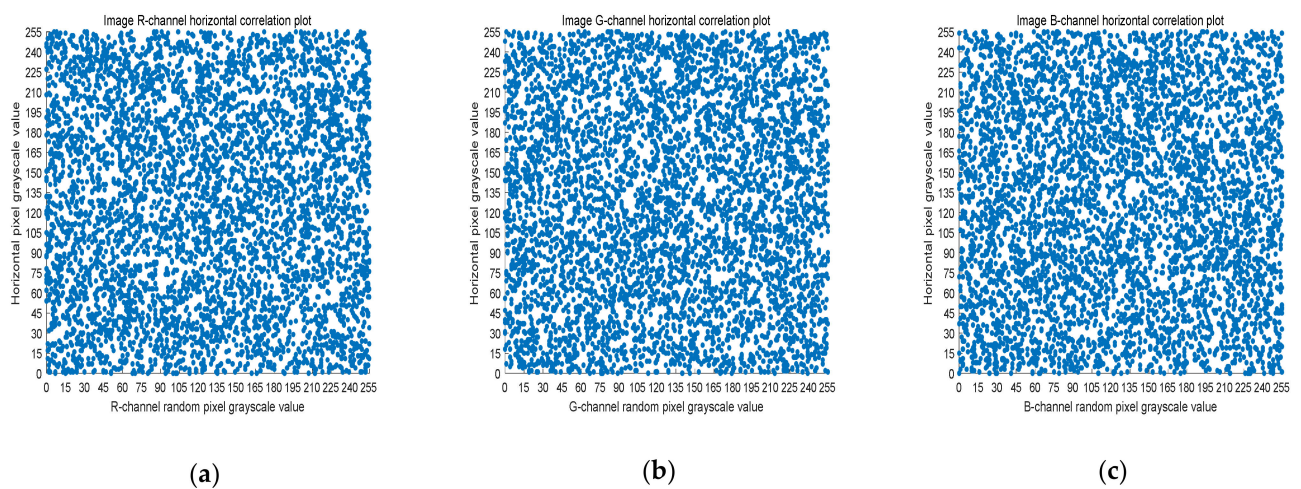


Figure 9. Scatter plot of horizontal RGB component of encrypted image. (a) R-channel scatter plot; (b) G-channel scatter plot; (c) B-channel scatter plot.

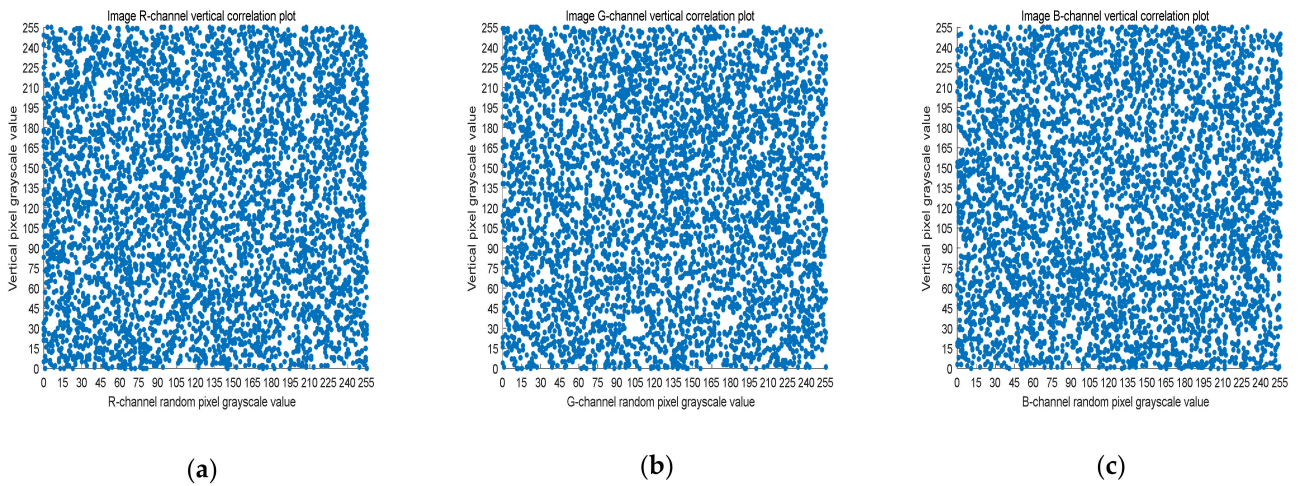


Figure 10. Scatter plot of vertical RGB component of encrypted image. (a) R-channel scatter plot; (b) G-channel scatter plot; (c) B-channel scatter plot.

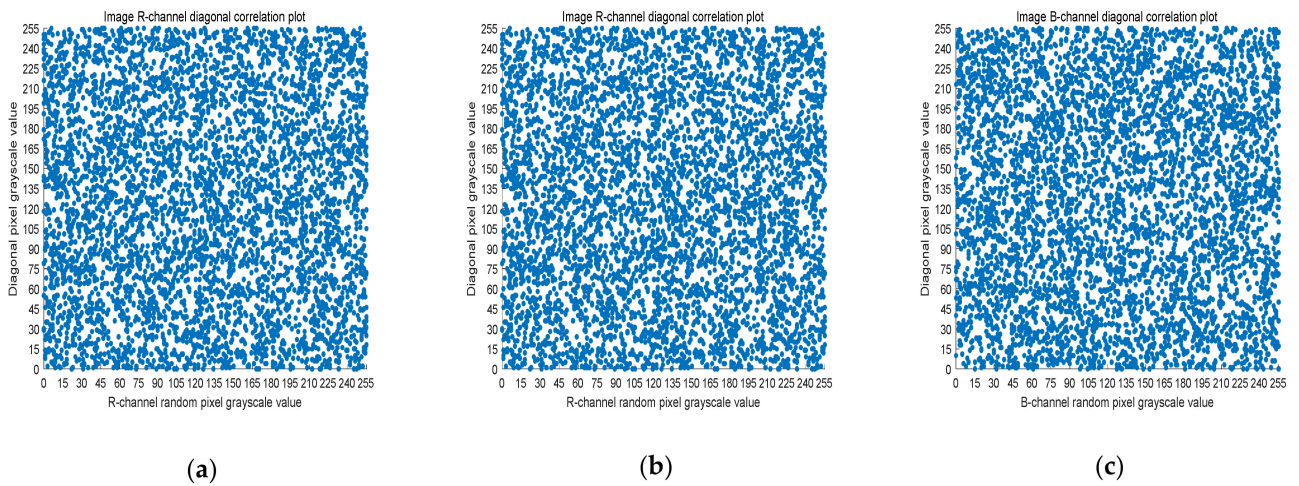


Figure 11. Scatter plot of diagonal RGB component of encrypted image. (a) R-channel scatter plot; (b) G-channel scatter plot; (c) B-channel scatter plot.

Table 2. Correlation of color image components in each direction.

		Horizontal Correlation	Vertical Correlation	Diagonal Correlation
Original image	R	0.97247	0.98656	0.96164
	G	0.97322	0.98675	0.96296
	B	0.94617	0.97208	0.92864
The encryption method proposed in this paper	R	−0.0096211	−0.011037	−0.00084143
	G	0.00096321	−0.0014868	−0.012429
	B	0.0022199	0.0015614	0.0045217
Literature [31]	R	0.0046	0.0046	0.0005
	G	0.0052	0.0058	0.0031
	B	0.0063	0.0084	0.0102
The large aerial images in the literature [33]	RGB average correlation	−0.0014	0.0039	−0.0027

4.4. Information Entropy

In this subsection, color images are used to test the information entropy. The information entropy reflects the degree of confusion of the pixel values of the whole image, and the higher the degree of confusion of the image, the higher the information entropy, and vice versa for the lower the information entropy. The information entropy is calculated by the following formula:

$$H = - \sum_{i=0}^L p(i) \log_2 p(i), \quad (9)$$

where L is the gray level of the image and $p(i)$ is the probability of the occurrence of the gray value i . From Table 3, the information entropy of the original Lena image and the information entropy of the encrypted image can be observed.

Table 3. Information entropy of color images.

Image	Information Entropy		
	R	G	B
original image	7.2682	7.5901	6.9951
Image encrypted in this article	7.9992	7.9993	7.9994
Literature [32]	7.9974	7.9971	7.9975
Literature [34]	7.9973	7.9972	7.9966

It can be observed from Table 3 that the results obtained by our method are close to the theoretical value of eight, which means that the cipher images will be more uniformly distributed and our encryption system is effective for encryption. In addition, according to the data in Table 3, it can be found that the encryption algorithm proposed in this paper has higher information entropy compared with other methods in the literature, which also shows that the encryption algorithm in this paper has a high degree of randomness.

5. Security Analysis

In this section, various known analytical methods are used to verify the security of our image encryption method.

5.1. Key Space Analysis

For every encryption system, the size of its key space is very important. The larger the key space is, the more it can prevent brute force attacks and increase the time cost of brute force attacks.

In the algorithm proposed in this paper, the key space consists of the following two parts: one is the key space of the Lorenz system and the other is the key space of Arnold mapping. The keys of the Lorenz system are $q_1, q_2, q_3, a_1, a_2, a_3, a'_1, a'_2, a'_3, x_0, y_0, z_0$; the keys of Arnold mapping are d, e, n . The system enters the chaotic state when $a'_1 + q_1 a_1 = 10$, $a'_2 + q_2 a_2 = 8/3$, $a'_3 + q_3 a_3 > 24.74$, introduced by the Lorenz system in Section 2.1. When $a'_3 + q_3 a_3 = 28$, the system enters the optimal chaotic state. So, the key of this chaotic system is chosen as close to the ideal value as possible. Where $q_1, q_2, q_3, a_1, a_2, a_3, a'_1, a'_2, a'_3, x_0, y_0, z_0$, the key space of the proposed encryption algorithm in this paper is much larger than 2^{100} if calculated with the computer precision of $10^{(-15)}$, and the key space of the superimposed Arnold mapping and the key space of the Lorenz chaos mapping make the key space much larger than the key space of the Lorenz chaos system, so it is known that the system can effectively prevent brute force attacks.

5.2. Key Sensitivity Analysis

Key sensitivity is an important measure of the security of an encryption algorithm. The more sensitive the key is to the value of the initial state of the system, the more secure

the algorithm is. Since the generation of chaotic sequences is closely related to each key, any change in the value of any key will lead to the generation of different chaotic sequences, so changing the value of any key can be used as a means to verify the key sensitivity; therefore, in this paper, the key a'_2 is chosen to verify the key sensitivity of this encryption algorithm. The main two aspects of verification are as follows: first, for the same plaintext image, the computer-processable accuracy is 16 bits, as concluded in the key space analysis in Section 5.1, and the smallest computer-processable accuracy is $10^{(-14)}$ for the key $a'_2 = 27.5$, so the other $\delta = 10^{(-14)}$ (the value of a'_2 is converted from 27.5 to $27.5 + (10^{(-14)})$) by an order of magnitude, to visually observe the difference between correctly decrypted and incorrectly decrypted encrypted images.

From Figure 12, it can be observed that any small change in the key cannot be decrypted correctly, which also shows that the encryption algorithm is key-sensitive.

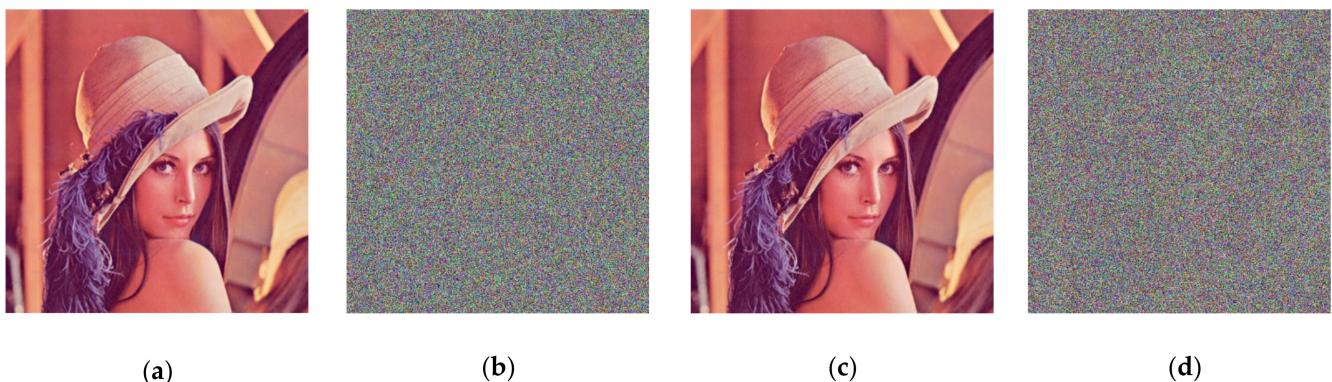


Figure 12. Key sensitivity test image: (a) Original picture I; (b) encrypted image; (c) decrypt image correctly; (d) decrypted image after small key change.

Second, compare the MSE (mean absolute error) values of the original image and the decrypted image to determine the key sensitivity. The mean square error is an IQA (image quality assessment) method, based on signal fidelity (or error signal sensitivity). The MSE is used to calculate the difference between a normal image and a cryptographic image. The degree of similarity between the decrypted image and the original image can be analyzed objectively, and the MAE is calculated as follows:

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N [f(i, j) - g(i, j)]^2, \quad (10)$$

where f and g are the normal image and its encrypted image, respectively, and M and N are the width and height of the image, respectively. MSE expresses the degree of distortion of an image, according to the statistical characteristics of the signal error, and a larger MSE indicates that the average squared error value of the two images at all pixel positions is larger, i.e., the more the distorted image deviates from the reference image, and the lower its degree of similarity. Conversely, the smaller the MSE value, the smaller the distortion and the higher the degree of similarity. In this paper, the average MSE of the image is used to test the key sensitivity of the image, and the change curve of the average mean square error between the decrypted image and the original image, corresponding to our encryption algorithm when the key a'_2 is varied between $\delta \in [-5 \times 10^{-9}, 5 \times 10^{-9}]$, is given in Figure 13. The two arrows on the left side of Figure 13 show the details of the change in the average MSE of the image between $-5 \times 10^{(-9)}$ and $-1 \times 10^{(-9)}$, and the two arrows on the right side of Figure 13 show the details of the change in the average MSE of the image between $-1 \times 10^{(-14)}$ and $1 \times 10^{(-14)}$, and the images show that when the key is changed very slightly, the average MSE of the decrypted image and the original image

will change dramatically, and the average MSE of the decrypted image and the original image after the key change is always above 8700.

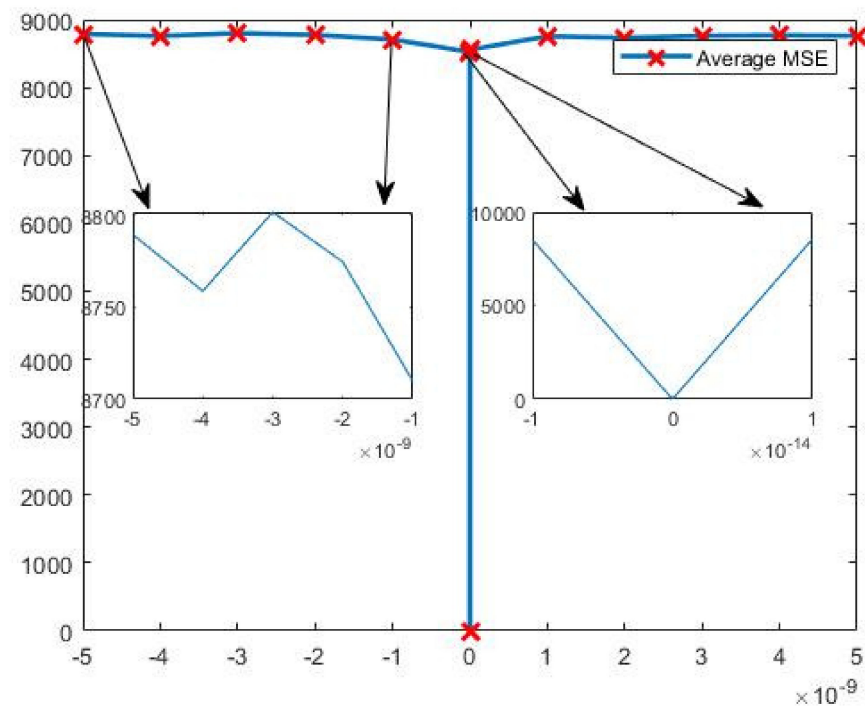


Figure 13. Average MSE corresponding to small change in key.

From Figure 13, we can observe that when the key a'_2 is correct, the MSE value of the decrypted image and the original image is zero, which indicates that the decrypted image is consistent with the original image. When the key a'_2 changes the $\delta = 10^{(-14)}$, the MSE changes greatly, which shows that the encryption algorithm proposed in this paper has very high key sensitivity. In Table 4, by comparing the MSE values of the original and encrypted images in the encryption algorithm proposed in this paper and the previously existing encryption algorithms, it is found that the MSE values of the encryption algorithm proposed in this paper are better. This also indicates that the confidentiality of the encryption algorithm proposed in this paper is higher.

Table 4. MSE values for different algorithms.

Algorithms	MSE Values
Methodology of this article	8932.0
Literature [35]	7775.0
Literature [33]	9875.5
AES	4600

5.3. Differential Attacks

To test the differential attack of our method, a number of color images were measured using the number of pixel change range (NPCR) and the uniform average change intensity (UACI), which are defined as follows:

$$\begin{cases} NPCR = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N D(i,j) \\ UACI = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \frac{|C1(i,j) - C2(i,j)|}{255} \end{cases} \quad (11)$$

where $D(i, j)$ is a Haveside function, and $D(i, j) = 0$ when $C1(i, j) = C2(i, j)$ and $D(i, j) = 1$ when $C1(i, j) \neq C2(i, j)$, according to the ranges provided in the literature [35]. First, we encrypt the original image to obtain a cryptographic image C1. Then, we choose a value and modify it in the original plain image, to obtain another cryptographic image C2. Finally, the NPCR and UACI are calculated by means of Equation (11). The simulation results are shown in Table 5. From Table 5, we can observe that when the image size is 512×512 , the NPCR value is greater than 0.996 and the UACI values are in the range of 0.33329–0.335541. Therefore, the proposed encryption system can pass a test according to the range proposed in the literature [43], which shows that our image encryption system can effectively resist differential attacks. Moreover, Table 5 shows the NPCR values and UACI values of different image encryption schemes. It can be observed, by Table 5, that these test results are within the acceptable interval, and our encryption method is closer to the theoretical values than other works.

Table 5. Performance against differential attacks.

Lena (512 × 512)	NPCR			UACI		
	R	G	B	R	G	B
Methodology of this article	0.99611	0.99627	0.99616	0.33400	0.33329	0.33483
Tiffany image in the literature [32]	0.9961	0.9961	0.9961	0.3626	0.3626	0.3626
Literature [44]	0.99602	0.99607	0.99601	0.334689	0.334965	0.334155
Literature [45]	0.99640	0.99633	0.99647	0.33488	0.33493	0.33509

5.4. Anti-Noise Capability Analysis

In this subsection, the color image Lena with a size of 512×512 is used as the original image, and will be used for encryption by our method. Pepper noise with different noise densities is added to the encrypted image, as shown in Figure 14, and the resulting decrypted image is shown in Figure 15. From these figures, it can be observed that the decrypted image can recover the original image information well when the pepper noise density changes from 0.0001 to 0.01, indicating that the algorithm has some resistance to noise attacks.

The higher the PSNR, the lower the distortion after compression, and the PSNR is the most common and widely used objective measurement to evaluate the image quality.

$$\text{PSNR} = 10 \cdot \lg\left(\frac{L^2}{\text{MSE}}\right) = 10 \cdot \lg\left(\frac{L^2}{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (f(i, j) - g(i, j))^2}\right), \quad (12)$$

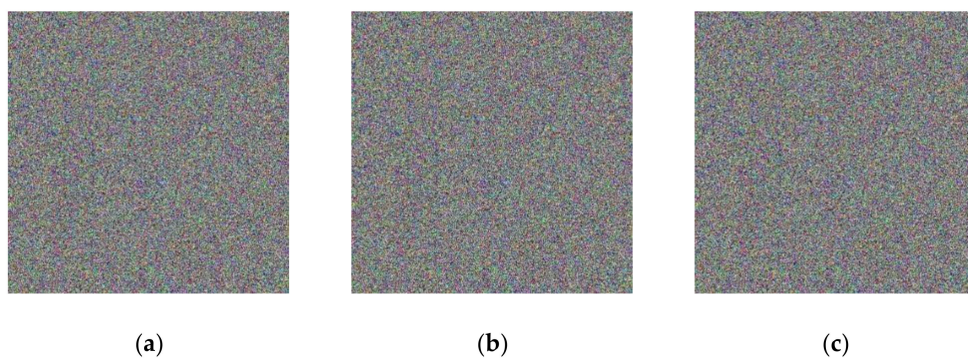


Figure 14. Encrypted image of different salt and pepper noise density: (a) salt and pepper noise density 0.0001; (b) salt and pepper noise density 0.001; (c) salt and pepper noise density 0.01.



Figure 15. Decryption image for different pretzel noise densities;(a) salt and pepper noise density 0.0001 decryption image; (b) salt and pepper noise density 0.001 decryption image; (c) salt and pepper noise density 0.01 decryption image.

According to Table 6, when the pepper noise density changes from 0.0001 to 0.01, the PSNR of the decrypted image and the original image gradually decreases, but its value is always greater than 27. Generally speaking, it is difficult to observe the distortion when the PSNR of the distorted image is above 35, which indicates the high quality of the image. When the PSNR value of the distorted image is between 28 and 35, the quality of the image will decrease, so the Figure 15, shows that the algorithm has some resistance to noise attack.

Table 6. PSNR corresponding to different pretzel noise decryption images.

Image (PSNR)	R	G	B
Pepper noise density 0.0001	48.0455	47.0210	50.5779
Pepper noise density 0.001	37.9455	38.6270	40.1578
Pepper noise density 0.01	27.6638	28.3126	29.5434

5.5. Analysis of Shear Resistance

To test the performance of the proposed encryption algorithm against clipping attacks, (a), (b), and (c) in Figure 16 show the encrypted Lena image (512×512) after some of the data are clipped off, and (a), (b), and (c) in Figure 17 show the decrypted images corresponding to Figure 16. It can be observed, from the decrypted images, that the cropped encrypted image can also be well recovered by the decryption algorithm, and, therefore, the proposed encryption algorithm has good performance against cropping aggressiveness.

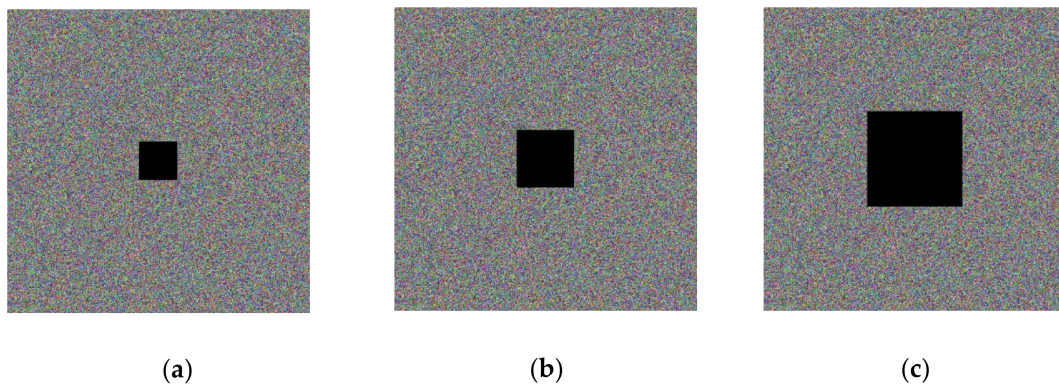


Figure 16. Data loss encryption image.



Figure 17. Data loss decryption image.

6. Conclusions

The algorithm used in this paper introduces the Lorenz system and Arnold mapping; it combines the plaintext key with the image and the key independent of the image as the parameters and initial values of the Lorenz system, so that the key space is greatly improved and the ciphertext is more random and resistant to attack, ensuring that the encrypted ciphertext is more secure. As the key acquisition is related to the plaintext, the key is acquired dynamically, and it is possible to achieve one key per image, which makes the proposed algorithm significantly more secure. Experiments have shown that the algorithm can not only encrypt the image effectively, but can also effectively prevent all kinds of attacks against the encrypted image, such as differential attack, shearing attack, noise attack, etc. Since the plaintext key acquisition in this paper is easier to implement, and the encryption does not decrease due to the reduction in system complexity, the image encryption system proposed in this paper will have better application prospects in the future. However, in the proposed image encryption algorithm, the complexity of the Lorenz system is high and takes up most of the time in encrypting the image information, so, in the future research process, we hope to find a chaotic system with better chaotic characteristics and lower complexity, to replace the Lorenz system, so that the proposed encryption algorithm can be more widely used in subsequent research.

Author Contributions: Z.W. conceived and wrote the paper. B.Z. and P.P. analyzed the data. B.Z. gave some theoretical guidance. C.S. gave some guidance for paper structure. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of China (No. 61801173).

Data Availability Statement: Data is contained within the article.

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

References

1. Matthews, R. On the derivation of a “Chaotic” encryption algorithm. *Cryptologia* **1989**, *8*, 29–41. [[CrossRef](#)]
2. Wang, Y.; Wong, K.W.; Liao, X.; Chen, G. A new chaos-based fast image encryption algorithm. *Appl. Soft Comput.* **2011**, *11*, 514–522. [[CrossRef](#)]
3. Sun, J. A chaotic image encryption algorithm combining 2D chaotic system and random XOR diffusion. *Phys. Scr.* **2021**, *96*, 105208. [[CrossRef](#)]
4. Zhang, J.; Chen, N.; Li, Y. Image encryption algorithm based on chaotic mapping and dynamic S-box. *J. Chin. Acad. Electron. Sci.* **2019**, *14*, 1129–1135.
5. Chen, S.; Tang, Y. Triple dislocation algorithm for RGB color images based on chaotic system. *J. Chongqing Univ. Posts Telecommun. Nat. Sci. Ed.* **2018**, *30*, 812–818.
6. Hu, Y.; Yu, S.; Zhang, Z. On the Security Analysis of a Hopfield Chaotic Neural Network-Based Image Encryption Algorithm. *Complexity* **2020**, *2020*, 2051653. [[CrossRef](#)]
7. Xu, X.; Chen, S. Single Neuronal Dynamical System in Self-Feedbacked Hopfield Networks and Its Application in Image Encryption. *Entropy* **2021**, *23*, 456. [[CrossRef](#)]

8. Chen, L.; Yin, H.; Huang, T.; Yuan, L.; Zheng, S.; Yin, L. Chaos in fractional-order discrete neural networks with application to image encryption. *Neural Netw.* **2020**, *125*, 174–184. [[CrossRef](#)] [[PubMed](#)]
9. Zhang, X.; Wang, X. Multiple-image encryption algorithm based on DNA encoding and chaotic system. *Multimed. Tools Appl.* **2018**, *78*, 7841–7869. [[CrossRef](#)]
10. Zhang, X.; Hu, Y. Multiple-image encryption algorithm based on the 3D scrambling model and dynamic DNA coding. *Opt. Laser Technol.* **2021**, *141*, 107073. [[CrossRef](#)]
11. Liu, Y.; Zhang, J. A Multidimensional Chaotic Image Encryption Algorithm based on DNA Coding. *Multimed. Tools Appl.* **2020**, *79*, 21579–21601. [[CrossRef](#)]
12. Zhu, J.; Ermann, N.; Chen, K.; Keyser, U.F. Image Encoding Using Multi-level DNA Barcodes with Nanopore Readout. *Small* **2021**, 2100711. [[CrossRef](#)] [[PubMed](#)]
13. Xie, D. Public Key Image Encryption Based on Compressed Sensing. *IEEE Access* **2019**, *7*, 131672–131680. [[CrossRef](#)]
14. Yamac, M.; Ahishali, M.; Passalis, N.; Raitoharju, J.; Sankur, B.; Gabbouj, M. Multi-Level Reversible Data Anonymization via Compressive Sensing and Data Hiding. *IEEE Trans. Inf. Forensics Secur.* **2020**, *16*, 1014–1028. [[CrossRef](#)]
15. Musanna, F.; Kumar, S. A novel image encryption algorithm using chaotic compressive sensing and nonlinear exponential function. *J. Inf. Secur. Appl.* **2020**, *54*, 102560. [[CrossRef](#)]
16. Dou, Y.; Li, M. An Image Encryption Algorithm Based on Compressive Sensing and M Sequence. *IEEE Access* **2020**, *8*, 220646–220657. [[CrossRef](#)]
17. Pourasad, Y.; Ranjbarzadeh, R.; Mardani, A. A New Algorithm for Digital Image Encryption Based on Chaos Theory. *Entropy* **2021**, *23*, 341. [[CrossRef](#)]
18. Shi, H.; Wang, L.D. A multi-process image encryption scheme based on compressed sensing and multidimensional chaotic systems. *J. Phys.* **2019**, *68*, 39–52.
19. Azam, N.A.; Hayat, U.; Ayub, M. A Substitution Box Generator, its Analysis, and Applications in Image Encryption. *Signal Process.* **2021**, *187*, 108144. [[CrossRef](#)]
20. Wang, Y.; Li, A.; Wang, S.; Wang, J.Y.; Hu, J.J. Image encryption algorithm based on the combination of CNN and traditional S-box. *Electro-Opt. Control* **2021**, *28*, 34–38, 57.
21. Yang, Y.G.; Guan, B.W.; Li, J.; Li, D.; Zhou, Y.H.; Shi, W.M. Image compression-encryption scheme based on fractional order hyper-chaotic systems combined with 2D compressed sensing and DNA encoding. *Opt. Laser Technol.* **2019**, *119*, 105661. [[CrossRef](#)]
22. Sun, C.; Wang, E.; Zhao, B. Image Encryption Scheme with Compressed Sensing Based on a New Six-Dimensional Non-Degenerate Discrete Hyperchaotic System and Plaintext-Related Scrambling. *Entropy* **2021**, *23*, 291. [[CrossRef](#)]
23. Wang, X.; Zhu, X.; Wu, X.; Zhang, Y. Image encryption algorithm based on multiple mixed hash functions and cyclic shift. *Opt. Lasers Eng.* **2018**, *107*, 370379. [[CrossRef](#)]
24. Wang, K.; Wu, X.; Wang, H.; Kan, H.; Kurths, J. New color image cryptosystem via SHA-512 and hybrid domain. *Multimed. Tools Appl.* **2021**, *80*, 18875–18899. [[CrossRef](#)]
25. Fu, C.; Zhang, G.-Y.; Zhu, M.; Chen, J.-X.; Lei, W.-M. A Fast Chaos-Based Colour Image Encryption Algorithm Using a Hash Function. *Informatica* **2018**, *29*, 651–673. [[CrossRef](#)]
26. Yang, Y.G.; Xu, P.; Yang, R.; Zhou, Y.H.; Shi, W.M. Quantum Hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption. *Sci. Rep.* **2016**, *6*, 19788. [[CrossRef](#)]
27. Zhou, N.; Chen, W.; Yan, X.; Wang, Y. Bit-level quantum color image encryption scheme with quantum cross-exchange operation and hyper-chaotic system. *Quantum Inf. Process.* **2018**, *17*, 137. [[CrossRef](#)]
28. Wang, Z.; Xu, M.; Zhang, Y. Review of Quantum Image Processing. *Arch. Comput. Methods Eng.* **2021**, 1–25. [[CrossRef](#)]
29. Butt, K.K.; Li, G.; Masood, F.; Khan, S. A Digital Image Confidentiality Scheme Based on Pseudo-Quantum Chaos and Lucas Sequence. *Entropy* **2020**, *22*, 1276. [[CrossRef](#)]
30. Plas-Garza, M.A.; Zambrano-Serrano, E.; Rodríguez-Cruz, J.R.; Posadas-Castillo, C. Implementation of an encrypted-compressed image wireless transmission scheme based on chaotic fractional-order systems. *Chin. J. Phys.* **2020**, *71*, 22–37. [[CrossRef](#)]
31. Wang, K.; Wu, X.; Gao, T. Double color images compression–encryption via compressive sensing. *Neural Comput. Appl.* **2021**, 1–22. [[CrossRef](#)]
32. Masood, F.; Ahmad, J.; Shah, S.A.; Jamal, S.S.; Hussain, I. A Novel Hybrid Secure Image Encryption Based on Julia Set of Fractals and 3D Lorenz Chaotic Map. *Entropy* **2020**, *22*, 274. [[CrossRef](#)]
33. Masood, F.; Boulila, W.; Ahmad, J.; Arshad, A.; Sankar, S.; Rubaiee, S.; Buchanan, W. A Novel Privacy Approach of Digital Aerial Images Based on Mersenne Twister Method with DNA Genetic Encoding and Chaos. *Remote. Sens.* **2020**, *12*, 1893. [[CrossRef](#)]
34. Chen, L.-P.; Yin, H.; Yuan, L.-G.; Lopes, A.M.; Machado, J.A.T.; Wu, R.-C. A novel color image encryption algorithm based on a fractional-order discrete chaotic neural network and DNA sequence operations. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 866–879. [[CrossRef](#)]
35. Ahmad, J.; Masood, F.; Shah, A.; Jamal, S.S.; Hussain, I. A Novel Secure Occupancy Monitoring Scheme Based on Multi-Chaos Mapping. *Symmetry* **2020**, *12*, 350. [[CrossRef](#)]
36. Gao, X. A color image encryption algorithm based on an improved Hénon map. *Phys. Scr.* **2021**, *96*, 065203. [[CrossRef](#)]
37. Munir, N.; Khan, M.; Hazzazi, M.M.; Aijaedi, A.; Ismail, A.A.K.H.; Alharbi, A.R.; Hussain, I. Cryptanalysis of Internet of Health Things Encryption Scheme Based on Chaotic Maps. *IEEE Access* **2021**, *9*, 105678–105685. [[CrossRef](#)]

38. Al-Hazaimeh, O.M.; Al-Jamal, M.F.; Alhindawi, N.; Omari, A. Image encryption algorithm based on Lorenz chaotic map with dynamic secret keys. *Neural Comput. Appl.* **2019**, *31*, 2395–2405. [[CrossRef](#)]
39. Ye, G.; Zhao, H.; Chai, H. Chaotic image encryption algorithm using wave-line permutation and block diffusion. *Nonlinear Dyn.* **2015**, *83*, 2067–2077. [[CrossRef](#)]
40. Yu, J.Z.; Vinc, T.L. Control study of chaotic Lorena systems. *J. Phys.* **1998**, *47*, 397–402.
41. Wu, C. Discrete Arnold transform improvement and its application in image dislocation encryption. *J. Phys.* **2014**, *63*, 91–110.
42. Li, C.; Luo, G.; Li, C. Image encryption scheme based on skew tent chaotic mapping and Arnold transform. *Comput. Appl. Res.* **2018**, *35*, 3424–3427.
43. Zhou, S.; Wang, X.; Wang, M.; Zhang, Y. Simple colour image cryptosystem with very high level of security. *Chaos Solitons Fractals* **2020**, *141*, 110225. [[CrossRef](#)]
44. Yildirim, M. A color image encryption scheme reducing the correlations between R, G, B components. *Optik* **2021**, *237*, 166728. [[CrossRef](#)]
45. Cheng, G.; Wang, C.; Chen, H. A Novel Color Image Encryption Algorithm Based on Hyperchaotic System and Permutation-Diffusion Architecture. *Int. J. Bifurc. Chaos* **2019**, *29*. [[CrossRef](#)]