

Article

A Chaos-Based Encryption Algorithm to Protect the Security of Digital Artwork Images

Li Shi ¹, Xiangjun Li ^{2,3,*}, Bingxue Jin ^{2,3,*} and Yingjie Li ^{2,3}¹ School of Art, Nanchang University, Nanchang 330031, China; shili@ncu.edu.cn² School of Software, Nanchang University, Nanchang 330031, China; 418000220008@email.ncu.edu.cn³ Jiangxi Provincial Key Laboratory of Data Security Technology, Nanchang 330031, China

* Correspondence: lixiangjun@ncu.edu.cn (X.L.); 408000220011@email.ncu.edu.cn (B.J.)

Abstract: Due to the security weaknesses of chaos-based pseudorandom number generators, in this paper, a new pseudorandom number generator (PRNG) based on mixing three-dimensional variables of a cat chaotic map is proposed. A uniformly distributed chaotic sequence by a logistic map is used in the mixing step. Both statistical tests and a security analysis indicate that our PRNG has good randomness and is more complex than any one-dimensional variable of a cat map. Furthermore, a new image encryption algorithm based on the chaotic PRNG is provided to protect the content of artwork images. The core of the algorithm is to use the sequence generated by the pseudorandom number generator to achieve the process of disruption and diffusion of the image pixels, so as to achieve the effect of obfuscation and encryption of the image content. Several security tests demonstrate that this image encryption algorithm has a high security level.

Keywords: chaos; pseudorandom number generator; artwork image encryption

MSC: 94-02



Citation: Shi, L.; Li, X.; Jin, B.; Li, Y. A Chaos-Based Encryption Algorithm to Protect the Security of Digital Artwork Images. *Mathematics* **2024**, *12*, 3162. <https://doi.org/10.3390/math12203162>

Academic Editor: Cheng-Chi Lee

Received: 2 September 2024

Revised: 23 September 2024

Accepted: 24 September 2024

Published: 10 October 2024



Copyright: © 2024 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

Pseudorandom number sequences have been widely used in many different kinds of scientific fields, e.g., spread-spectrum communication, Monte Carlo method in numerical simulation, and cryptography [1–3]. Owing to the security weakness of linear structure generators [4,5], a chaotic system, based on its highly nonlinearity, including sensitivity to initial values and parameters, unpredictability, and randomness, has been regarded as an important source for generating pseudorandom bit sequences. Until now, a large number of chaos-based PRNGs have been proposed [2,6–14].

Among all the chaos-based PRNGs, the PRNG based on a one-dimensional chaotic map has received the most attention for its easy implementation [2,6–9]. Kocarev et al. and Addabbo et al. both propose a chaotic random number generator based on a piecewise linear one-dimensional map [2,6]; Kanso and Smaoui propose two different kinds of PRNGs based on a single logistic map and two coupled logistic maps [7]; Cristina et al. compose a couple of tent maps with different parameters and use this new chaotic scheme for generating a PRNG [8]. However, studies show that the PRNG based on a one-dimensional chaotic map is not always secure enough as the authors claimed [15,16]. Some chaos theory methods are proven to be quite useful for extracting such information from chaotic systems, especially for a low-dimensional chaotic map. Almost all the low-dimensional chaotic systems can be successfully attacked by using the phase space reconstruction method as Short claimed in [10].

Therefore, using high-dimensional chaotic systems to generate PRNGs is an effective method to improve security flaws. Some different kinds of PRNGs based on high-dimensional chaotic systems are also proposed [11–14,17]. Liu analyzes the statistics and complexity of chaotic pseudorandom sequence generated by the Lorenz system [11]; Hu

et al. propose a PRNG based on high-dimensional Chen chaotic system [12]; Akhahani et al. use a three-dimensional quantum chaotic map for generating good pseudorandom numbers [14]; in [17], Zhao et al. propose a hyperchaotic algorithm that depends on a self-turbulent PRNG. For gaining a perturbation of different dimensions, a feedback controller in that algorithm was utilized to avoid the transient cycle phenomenon. Although high-dimensional chaotic systems always have dynamical complexity and are more capable to resist phase space reconstruction, the generated PRNGs are still not secure enough. This is because almost all the PRNGs based on high-dimensional chaotic systems are constructed solely on a one-dimensional variable. As we know, the complexity of any one-dimensional variable is always lower than the system complexity and the high complexity of high-dimensional chaotic systems will not guarantee a high complexity of each dimensional variable.

In this paper, we propose a pseudorandom number generator based on a 3D cat chaotic map and apply it to the encryption of art images and the contributions are as follows:

1. The multidimensional variables of the three-dimensional cat chaotic map are mixed, and the nonlinear interactions between different dimensions are exploited to generate a sequence of pseudorandom numbers with significantly higher complexity than a single-dimensional variable.
2. Pseudorandom number sequences can be used in many different types of domains such as image encryption [18–22]. Therefore, an image encryption algorithm based on the above complex pseudorandom number sequence is proposed. The algorithm utilizes the randomness and complexity of the pseudorandom number sequence to encrypt the digital artwork image, effectively securing the image content.

The necessity of adopting encryption algorithms for digital art images is self-evident. Firstly, at the level of copyright protection, digital artwork condenses the efforts and creativity of artists and its copyright should be respected and protected in the same way as traditional artwork. Encryption algorithms can effectively prevent works from being illegally copied, distributed, or tampered with; safeguard the legitimate rights and interests of artists; and stimulate their enthusiasm for continuous creation. Secondly, in terms of market value, the uniqueness and scarcity of digital artworks are key factors in market pricing. If security is not guaranteed, easy copying will flood the market, weakening its market value and damaging the interests of artists and investors. Furthermore, digital artworks, as treasures of modern culture, contain rich historical, cultural and artistic values. Ensuring their security through encryption is not only a respect for cultural heritage, but also a necessary move to leave precious cultural assets for future generations. In summary, the use of encryption algorithms to protect digital art images is a multiple need to maintain copyright, safeguard the market, and pass on culture.

The rest of this paper is organized as follows: Some brief introductions of the required chaotic maps and the generation scheme of our pseudorandom number sequences are proposed in Section 2. Randomness and security analyses of this PRNG are proposed in Section 3. In Section 4, we introduce a new artwork image encryption algorithm with some security analyses. Finally, Section 5 concludes the whole paper.

2. PRNG Based on Mixing Three-Dimensional Variables of Cat Chaotic Map

In this section, the construction process of a pseudorandom number generator based on a 3D cat chaotic map is presented.

2.1. Three-Dimensional Cat Map

The mathematical model of a three-dimensional cat map can be described as [23]

$$X_{i+1} = AX_i \bmod 1 \quad (1)$$

where $X_i = (x_i, y_i, z_i)$ is the three-dimensional state variable and A is a linear matrix and can be written as

$$A = \begin{pmatrix} 1 + a_x a_z b_y & a_z & a_y + a_x a_z + a_x a_y a_z b_y \\ b_z + a_x b_y + a_x a_z b_y b_z & 1 + a_z b_z & a_y a_z + a_x a_y a_z b_y b_z + a_x a_z b_z + a_x a_y b_y + a_x \\ a_x b_x b_y + b_y & b_x & 1 + a_x a_y b_x b_y + a_x b_x + a_y b_y \end{pmatrix} \quad (2)$$

As a special case, by simply setting $a_x = a_y = a_z = b_x = b_y = b_z = 1$, we have

$$A = \begin{pmatrix} 2 & 1 & 3 \\ 3 & 2 & 5 \\ 2 & 1 & 4 \end{pmatrix} \quad (3)$$

and the cat map can be simply modeled by

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 \\ 3 & 2 & 5 \\ 2 & 1 & 4 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \bmod 1 \quad (4)$$

In this paper, the parameters are set as above if not otherwise specified. For the constructed three-dimensional cat chaotic map, its xy -dimensional, xz -dimensional, and yz -dimensional state variables are shown in Figure 1a, 1b and 1c, respectively. Figure 1d shows the chaotic attractor map of this system, from which it can be seen that the three-dimensional state variables of Equation (4) are all uniformly distributed in the interval $(0, 1)$. And the maximum Lyapunov exponent of this map is calculated to be 0.654.

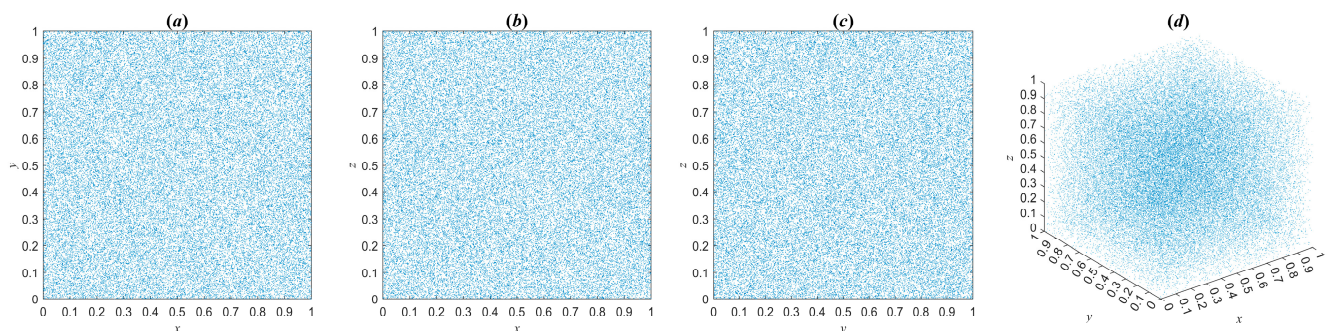


Figure 1. The state variables of cat chaotic map. (a) xy -dimensional; (b) xz -dimensional; (c) yz -dimensional; (d) chaotic attractor.

2.2. Logistic Map

The one-dimensional logistic map may be the most widely studied chaotic map. The zero-mean logistic map can be written as follows:

$$u_{i+1} = 1 - 2u_i^2 \quad (-1 \leq u_i \leq 1) \quad (5)$$

By setting an initial value u_0 , we generate a sequence denoted as $\{u_i\}$. As we know, the distribution of $\{u_i\}$ is not uniform. In [24], a method for generating uniform integer sequences by Equation (5) is proposed.

Divide the interval $I = [-1, 1]$ into N sub-intervals τ_i , $i = 0, 1, \dots, N - 1$. Denote $\tau_i = [t_i, t_{i+1})$, $i = 0, 1, \dots, N - 2$, and $\tau_{N-1} = [t_{N-1}, t_N]$, where

$$t_i = -\cos\left(\frac{i}{N}\pi\right) \quad (6)$$

Then, $\alpha = \{\tau_0, \tau_1, \dots, \tau_{N-1}\}$ is a finite measurable partition of I . Denote a correspondence $S: I \rightarrow \{0, 1, 2, \dots, N - 1\}$ from the set I to the set $\{0, 1, 2, \dots, N - 1\}$. For any $u \in I$, define $s(u) = i$, if $u \in \tau_i$. Here, $s(u)$ is the symbol representation of the real number u

according to the partition α . Then, the generated integer sequence is denoted as $\{s_i\}$ and can be proven to be uniformly distributed.

2.3. New PRNG Based on Mixing Three-Dimensional Variables of Cat Chaotic Map

Here, we propose a new PRNG based on mixing three-dimensional variables of a cat chaotic map. Taking $N = 3$, the PRNG is modeled as

$$b_i = \begin{cases} x_i, & \text{if } s_i = 0 \\ y_i, & \text{if } s_i = 1 \\ z_i, & \text{if } s_i = 2 \end{cases} \quad (7)$$

$$d_i = \text{floor}[b_i \cdot M]$$

where function $\text{floor}[x]$ denotes the maximum integer which is less than x and M is the cardinality of this PRNG. In the PRNG, the initial value of cat map (x_0, y_0 , and z_0) and the initial value of logistic map u_0 are used as the secret keys. The main frame of our PRNG is shown in Figure 2.

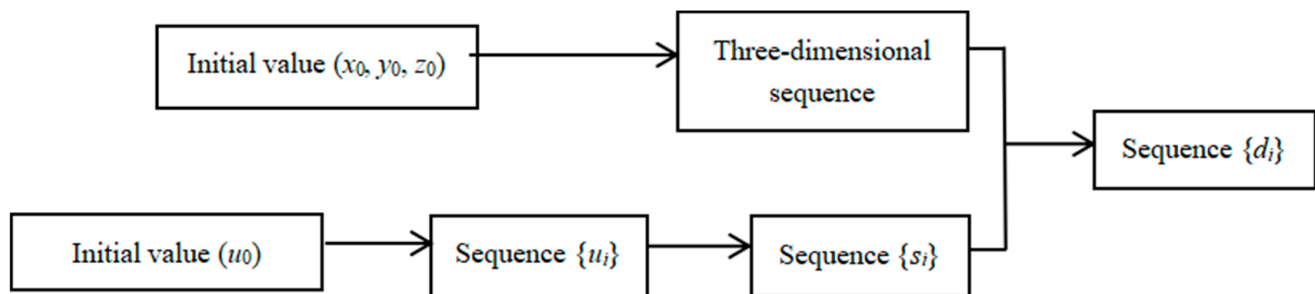


Figure 2. The main frame of our PRNG.

From Equation (7), we see that the generated sequence $\{b_i\}$ is based on mixing three-dimensional variables of a cat chaotic map, and sequence $\{d_i\}$ is just an integerization of $\{b_i\}$. Thus, the randomness and security of this PRNG are based on the properties of $\{b_i\}$.

Example 1. The sequence d_i is generated by the PRNG, so start by setting the initial values as follows: $x_0 = 0.2369$, $y_0 = 0.5758$, $z_0 = 0.6365$, $u_0 = 0.6$, and $M = 256$.

Step 1: Assign the initial values (x_0, y_0 , and z_0) to (x_1, y_1 , and z_1) to obtain $x_1 = 0.2369$, $y_1 = 0.5758$, and $z_1 = 0.6365$. Plug it into Equation (4) to obtain the corresponding sequence (x_i, y_i , and z_i).

$$x_1 = 0.2369, x_2 = 0.9591, x_3 = 0.7498, x_4 = 0.4807, x_5 = 0.3059, x_6 = 0.3128, \dots$$

$$y_1 = 0.5758, y_2 = 0.0448, y_3 = 0.9449, y_4 = 0.8662, y_5 = 0.3050, y_6 = 0.1877, \dots$$

$$z_1 = 0.6365, z_2 = 0.5956, z_3 = 0.3454, z_4 = 0.8261, z_5 = 0.1320, z_6 = 0.4448, \dots$$

Step 2: Then, substitute u_0 into Equation (5) to obtain the sequence u_i .

$$u_1 = 0.6, u_2 = 0.2800, u_3 = 0.8432, u_4 = -0.4220, u_5 = 0.6439, u_6 = 0.1708, \dots$$

Step 3: By combining Equation (6) and the mapping rules mentioned in Section 2.2, the sequence s_i is obtained.

$$s_1 = 2, s_2 = 1, s_3 = 2, s_4 = 1, s_5 = 2, s_6 = 1, \dots$$

Step 4: The sequence b_i is obtained by combining Equation (7) with the sequences (x_i, y_i , and z_i) and u_i .

$$b_1 = z_1 = 0.6365, b_2 = y_2 = 0.0448, b_3 = z_3 = 0.3454, b_4 = y_4 = 0.8662, b_5 = z_5 = 0.1320, b_6 = y_6 = 0.1877, \dots$$

Step 5: Then, using the relationship satisfied by b_i and d_i mentioned in Equation (7), the sequence d_i is generated and represented in binary.

$$d_1 = 162 = 10100010, d_2 = 11 = 00001011, d_3 = 88 = 01011000, d_4 = 221 = 11011101, d_5 = 33 = 00100001, d_6 = 48 = 00110000, \dots$$

Finally, the output sequence of the PRNG is obtained: 101000100000101101011000110111010010000100110000...

Next, we should compare the sequence $\{b_i\}$ to any one-dimensional variable of a cat map ($\{x_i\}$, $\{y_i\}$, and $\{z_i\}$), which proves that mixing all three-dimensional variables greatly improved the complex of any single dimensional variable.

Approximate entropy (ApEn) is used here to evaluate its complexity. ApEn was proposed by Pincus [25] and used to measure the probability of the new pattern generated in the sequences while the embedding dimension grows. Figure 3 shows the ApEn of sequences $\{b_i\}$, $\{x_i\}$, $\{y_i\}$, and $\{z_i\}$. From Figure 3, we know that the ApEn of sequence $\{b_i\}$ is obviously larger than the ApEn of $\{x_i\}$, $\{y_i\}$, and $\{z_i\}$, which means that the complexity is greatly improved.

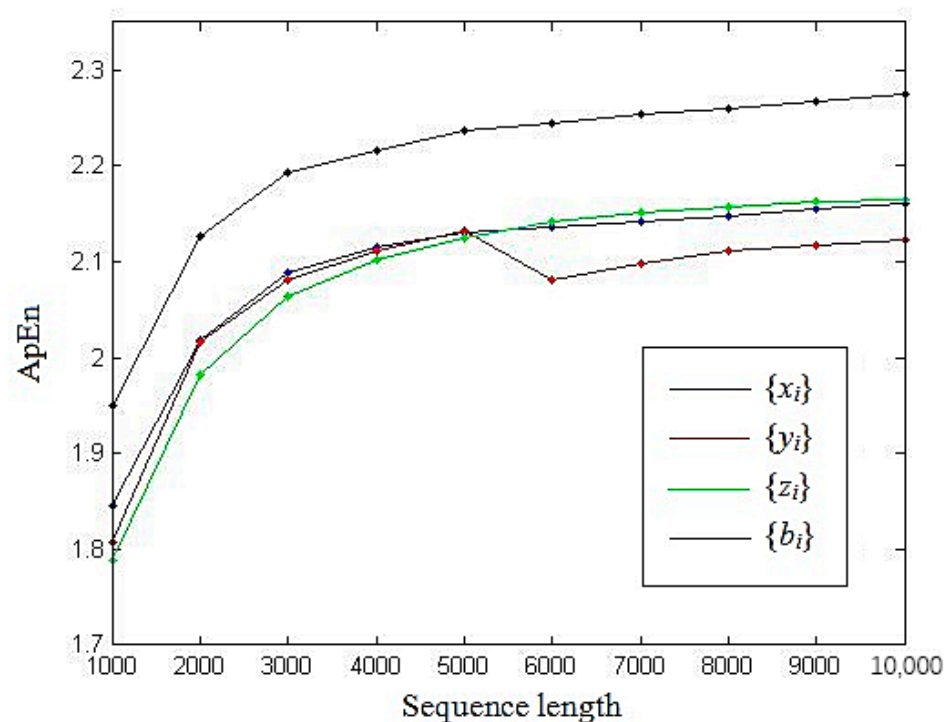


Figure 3. Comparison of ApEn for sequences $\{b_i\}$, $\{x_i\}$, $\{y_i\}$, and $\{z_i\}$.

3. Randomness and Security Analysis

In this Section, several tests are used to evaluate the randomness and security of our PRNG. Set $M = 2$ and the generated sequence $\{d_i\}$ correspond to a binary sequence.

3.1. Randomness Analysis

We used NIST statistical test suite [26] to evaluate the randomness of our PRNG. In all the randomness test suites, the NIST test suite may be the most popular one. The NIST tests suite is a statistical package comprising 16 tests that are developed to test the randomness of binary sequences. The significance level of each test was 0.01 in the NIST test. If $p\text{-value} \geq 0.01$, the test sequence is considered to pass the test. In this test, we randomly chose 500 groups of different initial values. Then, we generated 500 different bit sequences. The sequence length was 1,000,000. The results of NIST tests are shown in Table 1. From the passing ratio and mean value of p -values in Table 1, we see that the test sequences passed all the test suites, which means our PRNG has a good randomness.

Table 1. Results of NIST tests.

Test Index	Passing Ratio	Mean Value of p -Values
Approximate entropy	0.998	0.412479
Block frequency	0.996	0.169724
Cumulative sums	0.998	0.348731
FFT	0.994	0.531178
Frequency	0.998	0.235597
Linear complexity	0.996	0.365974
Random excursions	0.996	0.149732
Random excursion variant	0.996	0.266897
Longest runs of ones	0.996	0.433568
Overlapping template of all ones	0.998	0.138789
Rank	0.998	0.466872
Runs	0.988	0.394131
Serial	0.998	0.484836
Universal statistical	0.994	0.198424
Lempel–Ziv compression test	0.996	0.236567

3.2. Security Analysis

3.2.1. Key Space

In this PRNG, the initial values of cat map (x_0 , y_0 , and z_0) and the initial value of logistic map u_0 can be used as the secret keys. All these keys can be selected in the interval $(0, 1)$. Let the greatest accuracy be 10^{-14} ; then, the key space size can be approximately calculated as $10^{56} \approx 2^{186} > 2^{128}$, which is large enough to withstand brute-force attacks.

3.2.2. Key Sensitivity and Resistance to Differential Attacks

Differential attacks, or chosen-plaintext attacks, analyze the effects of a small difference in input keys on the difference of corresponding output sequences, which is proposed in [27]. Given two secret keys and their corresponding output sequences, a PRNG is said to resist differential attacks if the output sequences are completely different for a tiny difference on secret keys. In this test, we randomly chose 100 groups of initial values, changed each of the secret keys by only 10^{-14} , and compared the sequences by each bit. We denoted H to be the variance ratio by each bit. The results of key sensitivity for the secret keys x_0 , y_0 , z_0 , and u_0 are shown in Table 2. From Table 2, we can see that the variance ratios are all about 50% for every key secret, which means that our PRNG is extremely sensitive to all the initial values and can effectively resist to differential attacks.

Table 2. The variance ratio when x_0 is changed by 1 bit.

	Mean Value	Minimum Value	Maximum Value
$H(x_0)$	49.98%	49.88%	50.14%
$H(y_0)$	49.96%	49.86%	50.18%
$H(z_0)$	49.98%	49.89%	50.14%
$H(u_0)$	49.99%	49.90%	50.12%

3.2.3. Linear Complexity

Linear complexity is an important measure for a secure random sequence. For an ideal random binary sequence, the linear complexity should be close to the $n/2$ line. The linear complexity of our generated sequence is calculated and plotted in Figure 4. Figure 4 shows that the complexity line is extremely near the ideal $n/2$ line, which indicates that the generated sequence has high linear complexity.

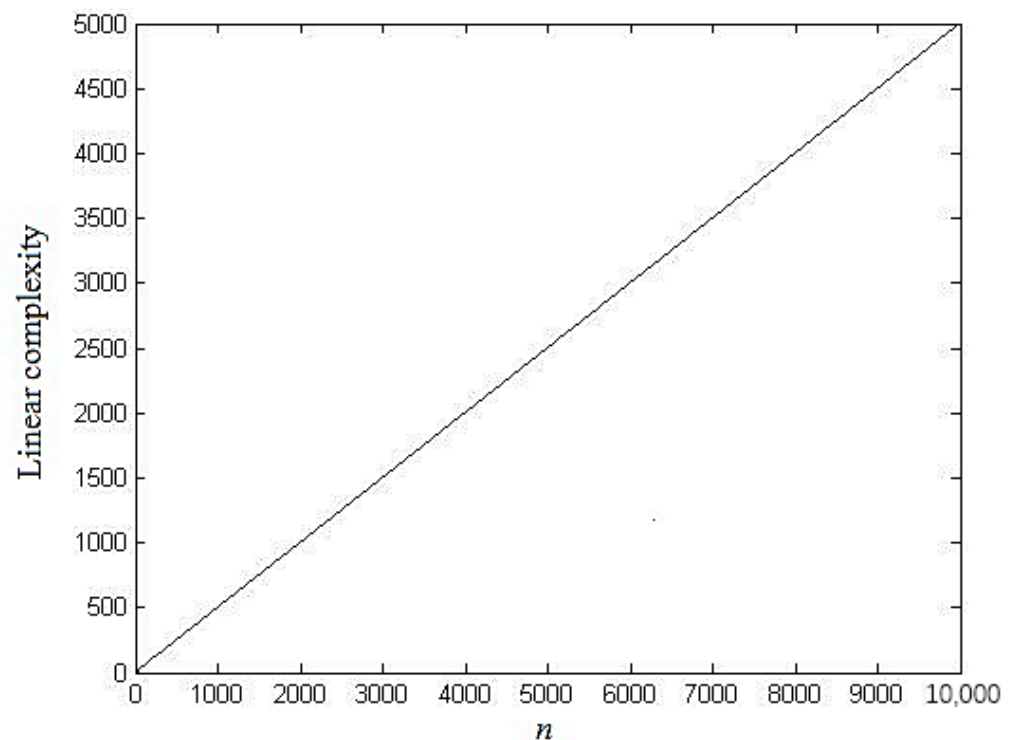


Figure 4. Linear complexity.

4. Application to Artwork Image Encryption

In this Section, we propose a new image encryption algorithm based on this PRNG. This encryption algorithm consists of shuffling and substitution algorithms. Some security tests are used to demonstrate the good performances of this algorithm.

4.1. Shuffling Algorithm

The shuffling algorithm is used to exchange the positions of pixels in the plain image and aims to reduce the correlation between adjacent pixels. We let $\{b_i\}$ be the output sequence by Equation (7) and let

$$k_i = b_i \times 10^8 \bmod 256 \quad (8)$$

Then, we obtained a number sequence $\{k_i\}$. Due to the uniformity of sequence $\{b_i\}$, $\{k_i\}$ can be ergodic quickly. We let $j(i)$ be the least integer which satisfies $k_{j(i)} = i$, $i = 0, 1, \dots, 255$. Rearranged, the sequence $\{k_{j(i)}\}$ is according to the order of $j(i)$. Then, we constructed the following matrix **B** according to sequence $\{k_{j(i)}\}$ as

$$B = B_{m,n} = \begin{cases} 1, & \text{if } m = i + 1, n = k_{j(i)}, i = 0, 1, \dots, 255 \\ 0, & \text{others} \end{cases} \quad (9)$$

Thus, there is only one 1 value in each row and column, while the others are 0. This matrix can be used to shuffle the pixel matrix of image. Let the pixel matrix be **S**; then, **BS** can be regarded as a row transformation of **S**, and **SB** can be regarded as the column transformation.

4.2. Substitution Algorithm

Substitution algorithm is used to change the pixel values. We let $M = 256$ in Equation (7) and constructed the corresponding chaotic matrix **C** by zig-zag scan of sequence $\{d_i\}$. Then, we substituted the pixel value of image **S** by using the following algorithm:

$$S^* = S + C \bmod 256 \quad (10)$$

Although the algorithm is simple, it is still secure due to the good randomness of **C**.

4.3. Image Encryption Algorithm

The detailed steps of our algorithm are described as follows:

Step 1: Read the plain image **S** and set encryption round T^* . Let $T = 0$.

Step 2: Choose a group of secret keys (x_0 , y_0 , z_0 , and u_0) and generate the pseudorandom sequences $\{b_i\}$ and $\{d_i\}$ according to Equation (8) by using the following initial values: $x_0 + \text{mean}\{S\}/1000 \bmod 1$, $y_0 + \text{mean}\{S\}/1000 \bmod 1$, $z_0 + \text{mean}\{S\}/1000 \bmod 1$, and $u_0 + \text{mean}\{S\}/1000 \bmod 1$.

Step 3: Generate the shuffling matrix **B** according to Section 4.1.

Step 4: Generate the substitution matrix **C** according to Section 4.2.

Step 5: Pre-multiply and post-multiply **S** by **B**.

Step 6: Substitute the pixel matrix by **C** according to Equation (10).

Step 7: $T = T + 1$.

Step 8: If $T = T^*$, go to step 9. Otherwise, go to step 2.

Step 9: Save as encrypted image S^* .

The decryption of the image is the inverse process of our encryption algorithm while generating the shuffling matrix and substitution matrix by using the same security keys. It should be noted that the mean value of pixel values of plain-text images should be transmitted securely to the receiver.

4.4. Experiment Tests and Security Analysis

In this section, three artwork images are used as the experiment images, and the encryption was set around $T^* = 3$. The numerical experiments are processed by Matlab 2014a on a computer with a 3.6 GHz CPU and an 8 GB memory.

4.4.1. Encryption and Decryption Experiments

The original images used for testing are referred to as Image 1, Image 2, and Image 3. The results of the encryption and decryption experiments are shown in Figure 5. Specifically, Figure 5a,d,g represent the images before encryption. Figure 5b,e,h represent the images after encryption. Figure 5c,f,i show the decrypted image. These images were transformed into an illegible noise form, from which the human eye cannot catch any clue of the original image, which fully demonstrates that the encryption algorithm we adopted imposed an effective protective barrier on the image content. Further, when we compared the decrypted images with the original images, they showed identical details and features, which verifies that our encryption and decryption processes is lossless.

4.4.2. Histogram Analysis

The histogram provides a visual response to the statistical characteristics of the image pixel data. When designing encryption algorithms, a key objective is to effectively destroy the original statistical characteristics of the image to enhance the security of the data. Figure 6 presents the significant differences in histogram performance between the original image and the encrypted image. Specifically, Figure 6a–c show the histograms of the three original images, which clearly show the peaks and distribution patterns specific to the respective images. However, in Figure 6d–f, the pixel distribution of the histograms of the corresponding encrypted images becomes extremely uniform. This comparison shows that our encryption algorithm successfully disrupted the statistical properties of the original images with extremely high security.



Figure 5. Encryption and decryption tests: (a,d,g) denotes the original image; (b,e,h) denotes the encrypted image; (c,f,i) denotes the decrypted image.

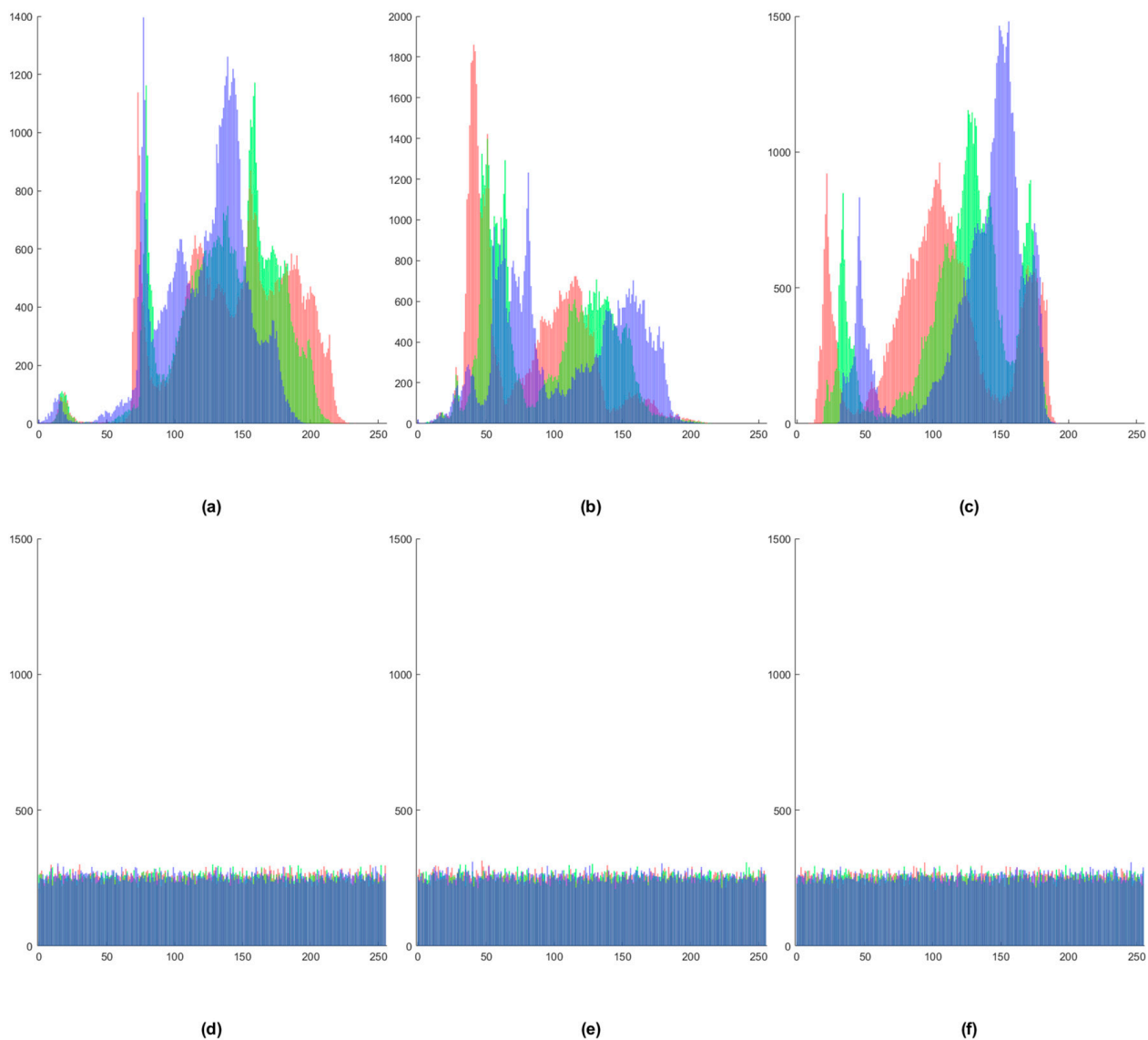


Figure 6. Histograms: (a–c) original images; (d–f) cryptographic images (the red, green and blue appearing in the diagram correspond to the three RGB colour channels).

4.4.3. Correlation Analysis

Given the intrinsic properties of image data, neighboring pixels often exhibit significant correlation; i.e., the value of one pixel can serve as a strong basis for predicting the values of its neighboring pixels. Therefore, one of the core objectives in designing efficient image encryption algorithms is to dismantle this inherent inter-pixel linkage and ensure that the pixels in the encrypted image no longer maintain strong interdependencies, thus enhancing the encryption effect.

And the inter-pixel correlation coefficient enables a quantitative assessment of the strength of correlation between neighboring pixels. Its calculation formula is shown below:

$$r = \frac{\sum_{i=1}^N \left(x_i - \frac{1}{N} \sum_{i=1}^N x_i \right) \left(y_i - \frac{1}{N} \sum_{i=1}^N y_i \right)}{\sqrt{\sum_{i=1}^N \left(x_i - \frac{1}{N} \sum_{i=1}^N x_i \right)^2 \times \sum_{i=1}^N \left(y_i - \frac{1}{N} \sum_{i=1}^N y_i \right)^2}} \quad (11)$$

where x_i and y_i denote the pixel values of a certain pair of neighboring pixels and N is the length of the randomly selected sequence. When the correlation coefficient is close to 1

or -1 , it indicates that there is a strong positive or negative correlation between the pixel pairs and when the correlation coefficient is close to 0, it indicates that there is almost no linear relationship between the pixel pairs; i.e., they are relatively independent.

Table 3 demonstrates the results of pixel correlation coefficients computed for the image before and after encryption on each color channel. Observing the data in the table, it is clear that before encryption, the correlation coefficient between image pixels is very close to 1, which indicates that there is a high degree of positive correlation between neighboring pixels in the original image in terms of color intensity. However, after the encryption algorithm, the pixel correlation coefficients of the image in each color channel drop dramatically and are almost infinitely close to 0. This significant change strongly proves that the encryption algorithm effectively destroys the original pixel-to-pixel correlation of the image, which makes the encrypted image statistically show a high degree of randomness and unpredictability.

Table 3. Correlation coefficient analysis.

Image	Direction	Plain Image			Cipher Image		
		R	G	B	R	G	B
Image 1	Horizontal	0.9577	0.9484	0.9329	0.0039	0.0028	−0.0127
	Vertical	0.9490	0.9381	0.9206	−0.0113	0.0145	−0.0018
	Diagonal	0.9233	0.9092	0.8831	−0.0070	−0.0035	0.0041
Image 2	Horizontal	0.9216	0.9253	0.9380	−0.0046	0.0021	0.0079
	Vertical	0.9625	0.9661	0.9719	0.0010	−0.0042	0.0046
	Diagonal	0.8972	0.9039	0.9192	−0.0008	0.0012	0.0047
Image 3	Horizontal	0.9811	0.9783	0.9769	0.0061	−0.0025	0.00045
	Vertical	0.9784	0.9758	0.9699	−0.0065	0.0047	−0.0127
	Diagonal	0.9682	0.9653	0.9586	−0.0034	0.0025	−0.0067

In addition, in order to demonstrate more intuitively the changes in the pixel correlation of the images before and after encryption, we plotted the corresponding pixel correlation maps, as shown in Figure 7. This atlas exhaustively presents a comparison of the pixel spatial layout between the original images (Image 1, Image 2, and Image 3) and the encrypted images in the red, green, and blue color channels, along different directions (horizontal, diagonal, and vertical). Among them, Figure 7a–j demonstrate the tight alignment and high correlation of the pairs of color component pixels in different directions in the original images. Figure 7j–r depict the pixel distribution states of the encrypted images in the same color channels and directions. The comparison reveals that the originally obvious pixel correlation is completely broken, and the pixel distribution becomes haphazard, making it almost impossible to recognize any regular pattern. This further validates the excellent performance of our proposed encryption algorithm in breaking image pixel correlation and enhancing data security.

4.4.4. Key Sensitive Analysis

When evaluating the security of an encryption scheme, key sensitivity, as one of the core metrics to measure its robustness, has an undeniable importance. An ideal encryption mechanism should ensure that it exhibits high sensitivity to any minor changes in the key, which is the key to preventing key guessing attacks and minor adjustment attacks. Specifically, even the slightest adjustment in the key should be enough to prevent the decryption process from correctly restoring the original plaintext image. In order to verify the performance of our designed encryption algorithm in terms of key sensitivity, we introduced a tiny perturbation of 2^{-12} to the key, and the test results are shown in Figure 8. From the figure, it can be visualized that the decrypted image obtained is wrong even though the key is only slightly changed. This shows that our encryption algorithm is extremely sensitive to the key.

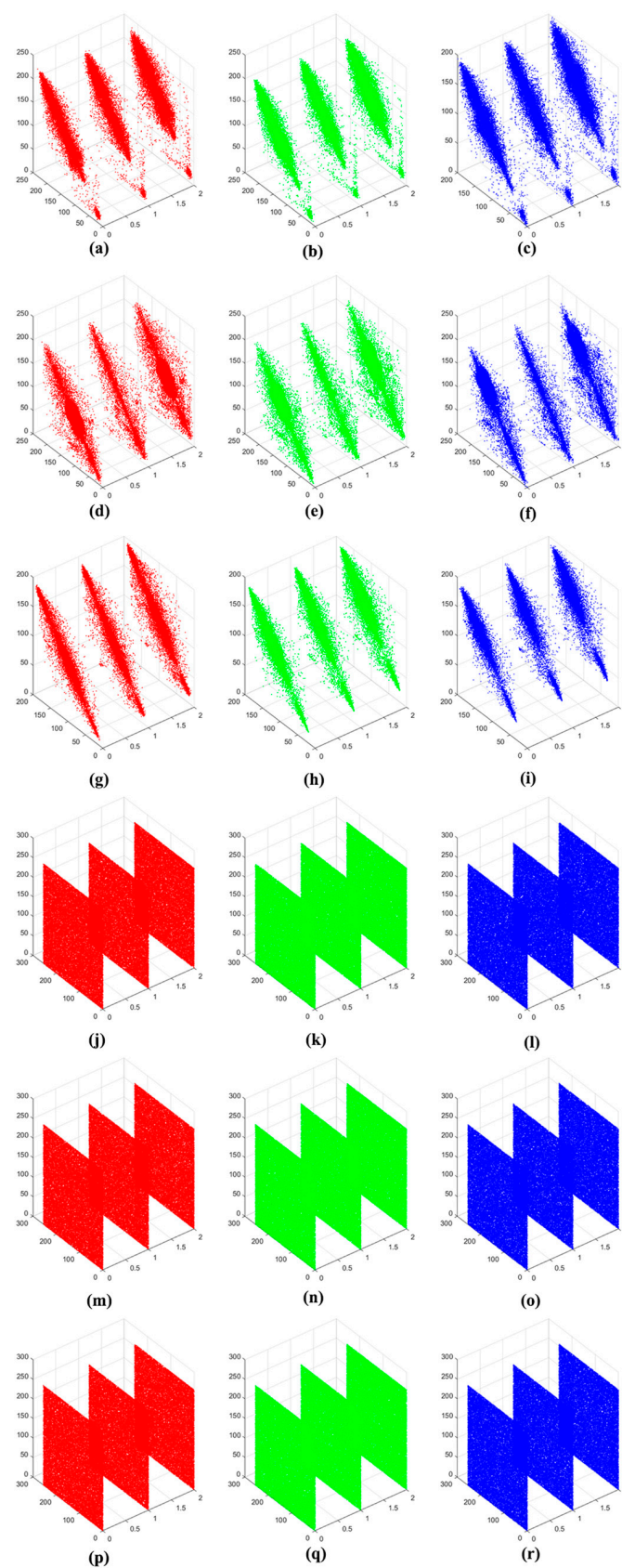


Figure 7. Pixel correlation analysis: (a–i) denote the pixel correlations of the original images of images 1–3 in different colour channels, horizontally, vertically and diagonally; (j–r) correspond to the pixel correlations of their secret images in different colour channels, horizontally, vertically and diagonally.

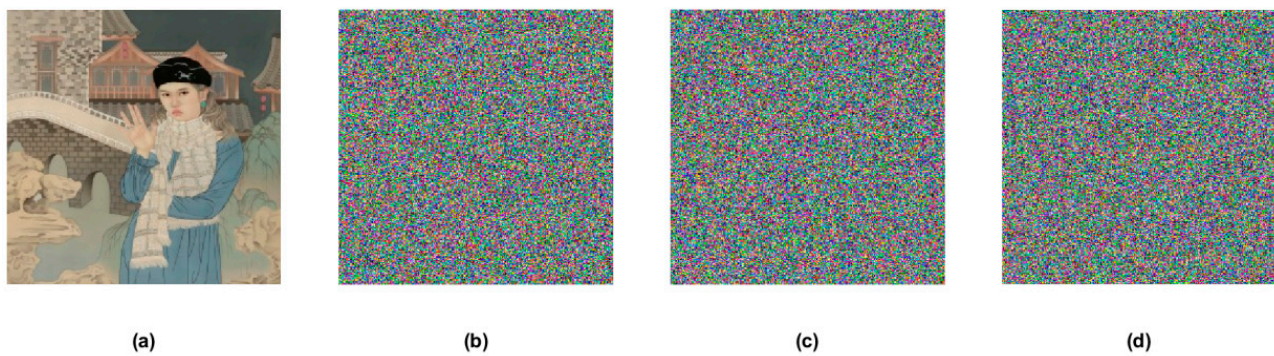


Figure 8. Key sensitive analysis: (a) correctly decrypted image; (b) $x_0 + 2^{-12}$; (c) $y_0 + 2^{-12}$; (d) $z_0 + 2^{-12}$.

4.4.5. Analysis of Information Entropy

The importance of information entropy as a widely adopted standard within the field of image encryption quality assessment and as a core metric for measuring the degree of disorder in a system cannot be overstated. It accurately quantifies the complexity and inherent uncertainty of the message content, and its calculation formula is as follows:

$$H(x) = -\sum_i^n e(x_i) \log_2 e(x_i) \quad (12)$$

n represents the total number of symbols; for image information, n is 8, and $e(x_i)$ denotes the probability of symbol x_i . We systematically recorded the information entropy values before and after encryption of Image 1, Image 2, and Image 3 and summarized these data in Table 4, aiming to visually demonstrate the effect of encryption operation on image randomness enhancement by quantitative means. Through comparative analyses, we found that the information entropy values of the encrypted images are all highly convergent to the ideal value 8, and the comparison with those in the literature is no less impressive. This significant result proves that the encryption process greatly enhances the randomness and unpredictability of the pixel arrangement of the image, ensuring the confidentiality and integrity of the information.

Table 4. Information entropy of different cipher images.

Image	Plain Image	Cipher Image
Image 1	7.0307	7.9971
Image 2	6.9860	7.9970
Image 3	6.8882	7.9972
Ref. [28]	7.1735	7.9022
Ref. [29]	7.2978	7.9860

4.4.6. Resistance to Differential Attack Analysis

Differential attacks are widely used to test the robustness and defense capability of encryption algorithms during the evaluation of image security. Such attacks mainly utilize the original plaintext data and their tiny variants as inputs, aiming at detecting predictable patterns latent in the output of the encryption algorithm and further breaking the encryption algorithm. In order to fully assess the ability of a cryptographic algorithm to resist such disparity attacks, we usually adopted a strategy of introducing a small change at the one-pixel level to the original input image, then encrypting the image before and after the change using a cryptographic algorithm, and finally measuring the difference between the two encrypted images. This process mainly relies on the calculation of two key metrics,

NPCR (Pixel Change Rate) and UACI (Uniform Average Change Intensity), as shown in Equations (13) and (15).

$$NPCR = \frac{\sum_{i=1}^M \sum_{j=1}^N D(i, j)}{M \times N} \times 100\% \quad (13)$$

where M and N are the width and height of the image, respectively, and $D(i, j)$ is a binary function defined as:

$$D(i, j) = \begin{cases} 1, & \text{if } C_1(i, j) \neq C_2(i, j) \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where $C_1(i, j)$ and $C_2(i, j)$ are the pixel values of the two encrypted images at position (i, j) , respectively.

$$UACI = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \frac{|C_1(i, j) - C_2(i, j)|}{255} \quad (15)$$

The experimental results are shown in Table 5, where it can be seen that the experimental NPCR and UACI values are extremely close to their ideal thresholds 0.9960 and 0.3355, which is not only a strong proof of the robustness of the algorithm, but also an intuitive manifestation of its strong defense capability against differential attacks. And comparing it to similar algorithms is no less impressive, further demonstrating the superiority of the algorithm.

Table 5. NPCR and UACI.

Image		NPCR	UACI
Image 1	R	0.9963	0.3340
	G	0.9960	0.3337
	B	0.9958	0.3339
Image 2	R	0.9961	0.3340
	G	0.9961	0.3348
	B	0.9962	0.3341
Image 3	R	0.9957	0.3363
	G	0.9959	0.3325
	B	0.9963	0.3351
Ref. [30]		0.9961	0.3350

4.4.7. Key Space Analysis

In measuring encryption algorithms, the size of the key space is a crucial factor that is directly related to the strength and security of the encryption algorithm. A larger key space means a larger number of possible keys and the more difficult it is for an attacker to crack an encrypted message by trying all possible keys. Typically, an algorithm is considered to be effective against exhaustive attacks when its key space is greater than 2^{128} . In this algorithm, our key is mainly composed of x_0 , y_0 , z_0 , and u_0 . When we set the computational precision to 10^{14} , its key space is $10^{56} \approx 2^{186}$, which is much larger than 2^{128} . This indicates that the algorithm effectively improves its ability to fight against exhaustive attacks and provides a solid guarantee for the safe transmission and storage of user data.

4.4.8. Analysis of Robustness

During the transmission of data, image data are often exposed to the risk of noise contamination and partial information loss, which poses a harsh robustness challenge to the design of image encryption algorithms. In order to comprehensively test the robustness of our proposed image encryption algorithm, we introduced salt and pepper noises as a simulation of the noise sources that may be encountered in the actual transmission process and applied 5% salt and pepper noises on the encrypted image. The experimental results show that in Figure 9a–c, despite the significant noise attack on the encrypted image, the quality of the decrypted image remains at a satisfactorily high level and both visual clarity

and information integrity are well preserved. This not only verifies the excellent ability of the encryption algorithm in coping with random noise, but also demonstrates its strong resistance to potential interfering factors during data transmission.

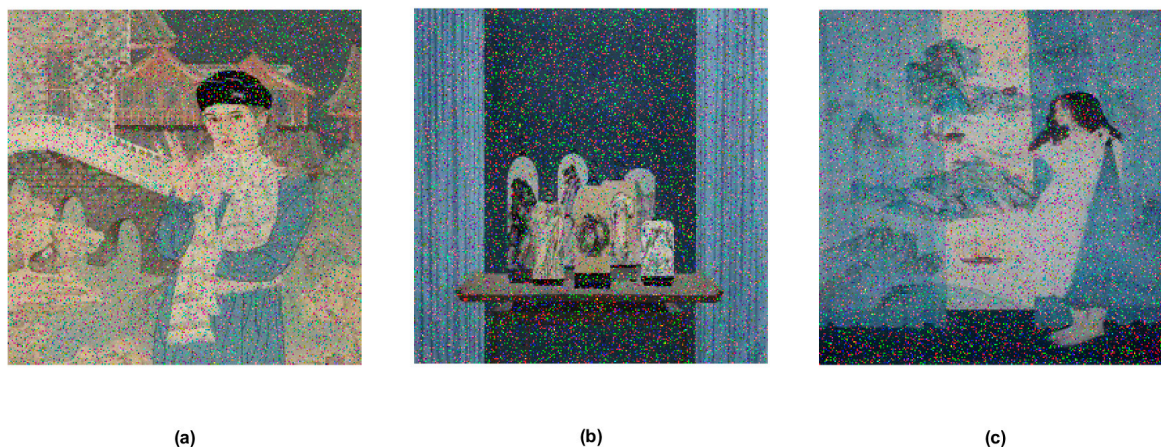


Figure 9. (a–c) denote the corresponding decrypted images after adding 5% salt and pepper noise to the three encrypted images, respectively.

4.4.9. Efficiency of Encryption and Decryption

For a cryptographic algorithm, ensuring security is crucial, but at the same time, efficiency is a factor that cannot be ignored. Runtime, as one of the important indicators for evaluating the performance of an algorithm, is directly related to the time and resource consumption required for the encryption process. Table 6 details the exact time required for the encryption and decryption processes when different sizes of images are compressed with different ratios using our proposed algorithm. This test aims to comprehensively evaluate the efficiency characteristics of the algorithm for different image sizes and compression ratios.

Table 6. Efficiency of encryption and decryption.

Algorithm	Encryption Time (s)	Decryption Time (s)	Speed (MB/s)
Image 1	0.1851	0.3029	0.3843
Image 2	0.1712	0.2166	0.4835
Image 3	0.1694	0.2122	0.4914

5. Conclusions

In this paper, after analyzing the security limitations in the existing research on pseudorandom number generators (PRNGs) based on chaos theory, a novel PRNG design based on the mixing of three-dimensional variables of cat chaotic maps is innovatively proposed. The design achieves unprecedented complexity and randomness in the mixing step by cleverly fusing uniformly distributed chaotic sequences of logic maps, effectively overcoming the security deficiencies of traditional PRNGs with one-dimensional chaotic variables. Statistical test results show that the PRNG proposed in this paper achieves an extremely high standard of randomness.

Further, this paper also designs and implements an image encryption algorithm based on chaotic PRNGs, especially for content protection of high-value data such as artwork images. The algorithm makes full use of the initial value sensitivity and long-term unpredictability of chaotic systems and combines with the unique characteristics of image data to construct an efficient and secure encryption mechanism. Through a series of tests such as histogram analysis, key sensitivity analysis, anti-differential attack analysis, the results show that the image encryption algorithm demonstrates a very high level of security, which provides strong security for image data such as artworks.

Although the cat chaotic map 3D variable mixing based PRNG proposed in this paper and its application in image encryption show significant advantages, there are still some limitations and challenges:

1. **Computational complexity:** Although the introduction of 3D chaotic systems significantly enhances randomness and security, it also increases computational complexity. In scenarios with high real-time requirements, such as real-time video encryption or high-speed data transmission encryption, this computational overhead may become a bottleneck.
2. **Hardware implementation difficulty:** It is technically challenging to effectively integrate the complex chaotic PRNG and its image encryption algorithms into hardware, such as FPGAs or ASICs, to achieve faster processing speeds and higher energy-efficiency ratios.
3. **Resistance to quantum attacks:** With the development of quantum computing technology, traditional encryption algorithms face the risk of being cracked. Although the chaotic system itself has some potential to resist quantum attacks, how to further enhance its quantum resistance to ensure that it remains secure in the era of quantum computing is an important topic for future research.

With the continuous development of information technology, PRNGs and their application in the field of security have a bright future. In the future, PRNGs based on chaos theory are expected to play an important role in the following aspects:

1. **Blockchain technology:** Consensus mechanism, cryptocurrency generation, smart contract execution, and other key aspects of blockchain require high-quality random numbers. The unpredictability and complexity of chaotic PRNGs make them ideal for random number generation in blockchain technology.
2. **Cloud computing and big data:** In cloud computing and big data environments, secure storage and transmission of data are crucial. Chaotic PRNGs can be used to build efficient encryption mechanisms to protect user privacy and data security.

Author Contributions: Validation, Y.L.; Writing—original draft, L.S.; Writing—review & editing, X.L. and B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62262023, 62262039, 62262040), the key project of Jiangxi Provincial Natural Science Foundation (20232ACB202009), the Jiangxi Provincial Key Laboratory of Data Security Technology (20242BCC32026), Key Research and Development Program of Jiangxi Province “Key Technologies Research and Demonstration Application of Security Control Platform for Financial Big Data Risk Control” (20243BBG71035), the Finance Science and Technology Special “Contract System” Project of Jiangxi Province (Grant Nos. ZBG20230418001, ZBG20230418014), the Market Supervision Administration Science and Technology Project of Jiangxi Province (Grant No.GSJK202305), the Jiangxi Province Educational Reform Key Project (JXJG-2020-1-2), and the Science and Technology Research Support Project of Jiangxi Provincial Education Department (GJJ2210701).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Kalouptsidis, N. *Signal Processing Systems, in Telecommunications and Signal Processing Series*; Wiley: New York, NY, USA, 1996.
2. Addabbo, T.; Alioto, M.; Fort, A.; Rocchi, S. A feedback strategy to improve the entropy of a chaos-based random bit generator. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2006**, *53*, 326–337. [\[CrossRef\]](#)
3. Alvarez, G.; Montoya, F.; Romera, M.; Pasto, G. Keystream cryptanalysis of a chaotic cryptographic method. *Comput. Phys. Commun.* **2003**, *156*, 205–207. [\[CrossRef\]](#)
4. Courtois, N.T.; Meier, W. Algebraic attack on stream ciphers with linear feedback. In *Advances in Cryptology—EUROCRYPT 2003, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, 4–8 May 2003*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 345–359.

5. Armknecht, F. Improving fast algebraic attacks. In *Fast Software Encryption, Proceedings of the 11th International Workshop, FSE 2004, Delhi, India, 5–7 February 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 65–82.
6. Stojanovski, T.; Kocarev, L. Chaos-based random number generators—Part I: Analysis. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2001**, *48*, 281–288. [\[CrossRef\]](#)
7. Kansa, A.; Smaoui, N. Logistic chaotic maps for binary numbers generations. *Chaos Solitons Fractals* **2009**, *40*, 2557–2568. [\[CrossRef\]](#)
8. Cristina, D.A.; Radu, B.; Ciprian, R. A new pseudorandom bit generator using compounded chaotic tent maps. In *Proceedings of the 2012 9th International Conference on Communications, Bucharest, Romania, 21–23 June 2012*; pp. 339–342.
9. Kumar, A.; Alzabut, J.; Kumari, S.; Rani, M.; Chugh, R. Dynamical properties of a novel one dimensional chaotic map. *Math. Biosci. Eng.* **2022**, *19*, 2489–2505. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Short, K.M. Steps toward unmasking secure communications. *Int. J. Bifurc. Chaos* **1994**, *4*, 959–977. [\[CrossRef\]](#)
11. Liu, N. Pseudo-randomness and complexity of binary sequences generated by the chaotic system. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 761–768.
12. Hu, H.P.; Liu, L.F.; Ding, N.D. Pseudorandom sequence generator based on the Chen chaotic system. *Comput. Phys. Commun.* **2013**, *184*, 765–768. [\[CrossRef\]](#)
13. Erkan, U.; Toktas, A.; Lai, Q. Design of two dimensional hyperchaotic system through optimization benchmark function. *Chaos Solitons Fractals* **2023**, *167*, 113032. [\[CrossRef\]](#)
14. Akhshani, A.; Akhavan, A.; Mobaraki, A.; Lim, S.C.; Hassan, Z. Pseudo random number generator based on quantum chaotic map. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 101–111. [\[CrossRef\]](#)
15. Udaltsov, V.S.; Goedgebuer, J.P.; Larger, L.; Cuenot, J.B.; Levy, P.; Rhodes, W.T. Cracking chaos-based encryption system ruled by nonlinear time delay differential equations. *Phys. Lett. A* **2003**, *308*, 54–60. [\[CrossRef\]](#)
16. Udaltsov, V.S.; Larger, L.; Goedgebuer, J.P. Time delay identification in chaotic cryptosystems ruled by delay-differential equations. *J. Opt. Technol.* **2005**, *72*, 373–377. [\[CrossRef\]](#)
17. Zhao, Y.; Gao, C.; Liu, J.; Dong, S. A self-perturbed pseudo-random sequence generator based on hyperchaos. *Chaos Solitons Fractals X* **2019**, *4*, 100023. [\[CrossRef\]](#)
18. Zhu, Y.; Wang, C.; Sun, J.; Yu, F. A chaotic image encryption method based on the artificial fish swarms algorithm and the DNA coding. *Mathematics* **2023**, *11*, 767. [\[CrossRef\]](#)
19. Feng, W.; Wang, Q.; Liu, H.; Ren, Y.; Zhang, J.; Zhang, S.; Qian, K.; Wen, H. Exploiting newly designed fractional-order 3D Lorenz chaotic system and 2D discrete polynomial hyper-chaotic map for high-performance multi-image encryption. *Fractal Fract.* **2023**, *7*, 887. [\[CrossRef\]](#)
20. Liu, Z.; Guo, Q.; Xu, L.; Ahmad, M.A.; Liu, S. Double image encryption by using iterative random binary encoding in gyrator domains. *Opt. Express* **2021**, *18*, 12033–12043. [\[CrossRef\]](#)
21. Singh, R.K.; Kumar, B.; Shaw, D.K.; Khan, D.A. Level by level image compression-encryption algorithm based on quantum chaos map. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 844–851. [\[CrossRef\]](#)
22. Kang, X.; Tao, R. Color image encryption using pixel scrambling operator and reality-preserving MPFRHT. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 1919–1932. [\[CrossRef\]](#)
23. Chen, G.R.; Mao, Y.B.; Chui, C.K. A symmetric image encryption scheme based on 3D chaotic cat map. *Chaos Solitons Fractals* **2004**, *21*, 749–761. [\[CrossRef\]](#)
24. Hu, H.P.; Liu, S.H.; Wang, Z.X.; Wu, X.G. A chaotic poly-phase pseudorandom sequence. *Acta Math. Sci.* **2004**, *24*, 251–256. (In Chinese)
25. Pincus, S.M. Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. USA* **1991**, *88*, 2297–2301. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Bassham, L.E.; Rukhin, A.L.; Soto, J.; Nechvatal, J.R.; Smid, M.E.; Leigh, S.D.; Levenson, M.; Vangel, M.; Heckert, N.A.; Banks, D.L. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; Special Publication (NIST SP); National Institute of Standards and Technology: Gaithersburg, MD, USA, 2010.
27. Biham, E.; Shamir, A. *Differential Cryptanalysis of the Data Encryption Standard*; Springer: London, UK, 1993.
28. Yildirim, M. Optical color image encryption scheme with a novel DNA encoding algorithm based on a chaotic circuit. *Chaos Solitons Fractals* **2022**, *155*, 111631. [\[CrossRef\]](#)
29. Lei, R.Q.; Liu, L.F. A two-dimensional chaotic model and its application in image encryption. *Phys. Scr.* **2024**, *99*, 075202. [\[CrossRef\]](#)
30. Xuejing, K.; Zihui, G. A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. *Signal Process. Image Commun.* **2020**, *80*, 115670. [\[CrossRef\]](#)

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