applied sciences

MDPI

Article

A New Chaotic Color Image Encryption Algorithm Based on Memristor Model and Random Hybrid Transforms

Yexia Yao [1], Xuemei Xu [1,*] and Zhaohui Jiang [2]

[1] School of Electronic Information, Central South University, Changsha 410083, China; yexia.yao@csu.edu.cn
[2] School of Automation, Central South University, Changsha 410083, China; jzh0903@csu.edu.cn
* Correspondence: xxm999@csu.edu.cn

Abstract: This paper skillfully incorporates the memristor model into a chaotic system, creating a two-dimensional (2D) hyperchaotic map. The system's exceptional chaotic performance is verified through methods such as phase diagrams, bifurcation diagrams, and Lyapunov exponential spectrum. Additionally, a universal framework corresponding to the chaotic system is proposed. To enhance encryption security, the pixel values of the image are preprocessed, and a hash function is used to generate a hash value, which is then incorporated into the secret keys generation process. Existing algorithms typically encrypt the three channels of a color image separately or perform encryption only at the pixel level, resulting in certain limitations in encryption effectiveness. To address this, this paper proposes a novel encryption algorithm based on 2D hyperchaotic maps that extends from single-channel encryption to multi-channel encryption (SEME-TDHM). The SEME-TDHM algorithm combines single-channel and multi-channel random scrambling, followed by local cross-diffusion of pixel values across different planes. By integrating both pixel-level and bit-level diffusion, the randomness of the image information distribution is significantly increased. Finally, the diffusion matrix is decomposed and restored to generate the encrypted color image. Simulation results and comparative analyses demonstrate that the SEME-TDHM algorithm outperforms existing algorithms in terms of encryption effectiveness. The encrypted image maintains a stable information entropy around 7.999, with average NPCR and UACI values close to the ideal benchmarks of 99.6169% and 33.4623%, respectively, further affirming its outstanding encryption effectiveness. Additionally, the histogram of the encrypted image shows a uniform distribution, and the correlation coefficient is nearly zero. These findings indicate that the SEME-TDHM algorithm successfully encrypts color images, providing strong security and practical utility.

Keywords: color image; image encryption; memristor model; hybrid transforms

check for updates

Academic Editors: Maria Pia Morigi and Marco Seracini

Received: 25 December 2024
Revised: 14 January 2025
Accepted: 15 January 2025
Published: 17 January 2025

Citation: Yao, Y.; Xu, X.; Jiang, Z. A New Chaotic Color Image Encryption Algorithm Based on Memristor Model and Random Hybrid Transforms. *Appl. Sci.* **2025**, *15*, 913. https://doi.org/10.3390/app15020913

Copyright: © 2025 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 rapid development of information technology, digital images have become increasingly important in modern communication as an intuitive and efficient means of information transmission [1]. Consequently, the demand for secure transmission of digital images is also increasing. To ensure the security and confidentiality of images, technologies such as data hiding, digital watermarking, and image encryption have emerged [2]. Among these, image encryption is a crucial algorithm for protecting image information security. Since color images contain a wider range of information and are more prevalent than grayscale images, the encryption of color images has become a hot research area [3,4].

The complex dynamical behavior exhibited by chaotic systems includes sensitivity to initial conditions, pseudo-randomness, and ergodicity. These features make chaotic

systems important in various fields, such as cryptography, data compression, and image processing [5–7]. Systems with larger chaotic ranges are less susceptible to external attacks, and the chaotic properties are more stable, which can be used to encrypt images [8]. As a nonlinear component, the memristor has attracted much attention since its introduction due to its potential to construct chaotic systems with more complex and diverse dynamics [9,10]. In this context, a large number of studies on chaotic systems based on memristors have emerged [11–13]. However, due to the insufficient complexity of the chaotic behavior in most systems, its application in image encryption is not very satisfactory [14–17]. Therefore, a 2D hyperchaotic model with complex dynamical behavior based on a memristor model is proposed in this paper.

In addition to chaotic systems, encryption algorithms are also particularly important. Previous studies have shown that many image encryption algorithms based on chaotic systems have some limitations, such as incomplete chaos, insufficient key space and so on. Wei Feng et al. proposes a multi-image encryption algorithm based on a novel fractional-order three-dimensional Lorenz chaotic system and a two-dimensional discrete polynomial hyperchaotic map [18]. However, the instability in the UACI and NPCR values indicates that the encryption algorithm's sensitivity to input changes is insufficient. As a result, the security of the algorithm is unstable and may be vulnerable to potential attacks. Jiuyang Xin et al. presented a novel Rubik's Cube-like permutation method based on a novel cryptosystem (3DVSCS-IES) [19]. However, its scrambling is not thorough enough, leading to its weak resistance to cropping. Yaning Li et al. presented a new color image encryption method by combining 2D chaotic map and compressed sensing [20]. Nevertheless, the inadequate key space used in encryption leads to its susceptibility to statistical attacks. Jianeng Tang et al. constructs a new two-dimensional chaotic map based on the interleaving of cosine and sine functions, utilizing two independent trigonometric functions without the need for composite operations [21]. While the chaotic map has a simple structure, its relatively low Lyapunov exponents suggest weak chaotic dynamics, which could impact its effectiveness in applications like secure communication. Based on the above problems, this paper proposes the SEME-TDHM algorithm that combines single-channel stochastic processing with multi-channel stochastic processing based on chaotic systems with complex dynamical behavior.

The subsequent section outlines the contributions and innovations of this paper:

(1) This paper presents a 2D hyperchaotic map integrated with a memristor model. The map exhibits complex chaotic behavior and offers a wide range of parameter choices. A framework based on this map is proposed, enabling the exploration of additional chaotic maps with superior performance and laying the groundwork for further studies on similar systems.

(2) A new chaotic color image encryption algorithm based on single channel random encryption to multi-channel random encryption named SEME-TDHM algorithm is designed. The randomness of the original image is enhanced by preprocessing to improve the security of the hash algorithm.

(3) A novel multi-channel random ranking algorithm is proposed, which randomly adjusts the obfuscated index sequence before each ranking. Additionally, a new local diffusion algorithm is introduced, using a localization diffusion matrix to diffuse pixel channels in multiple directions, disrupting their internal connections.

The structure of this article is as follows: Section 2 introduces a model of the 2D hyperchaotic map and analyzes its universal structure. Section 3 examines the chaotic behavior and performance of the proposed 2D hyperchaotic map. In Section 4, the SEME-TDHM algorithm for color images based on the chaotic map is presented. Section 5 discusses the experimental results, followed by a summary in Section 6.

2. 2D Hyperchaotic Map

An electric field-controlled memristor uses an electric field to regulate its resistance value. Typically, it consists of a dielectric layer and two electrodes. When the size of the electric field changes, the degree of polarization in the dielectric also changes, which causes the resistance value of the memristor to change.

$$I = I_0 \times \exp(-\frac{g}{g_0}) \times \sinh(\frac{V}{V_0}) \tag{1}$$

where V is the voltage, I is the current, and g denotes the gap between the upper and lower electrodes.

In a compact model, the oxide layer is positioned between the two electrodes, and the resistance strongly correlates with the gap distance (denoted as g) between them. The average distance g is used to determine the resistance of the RRAM through the electron tunneling conduction mechanism, resulting in an exponential increase in resistance with gap distance [22]. When a voltage (V) is applied to the memristor, a non-linear relationship between the resistance value and the applied voltage emerges. The resistance value decreases, remains stable, or increases with the voltage, and the current changes accordingly. Consequently, the current-voltage (I–V) relationships of the RRAM model are expressed as Equation (1). As nonlinearity is one of the key factors that cause chaos, it is considered to incorporate the memristor model into the chaotic system.

The inherent nonlinearity, complexity, and randomness of this memristor model used in the chaotic model building usually help to generate chaotic behavior, which can improve security and randomness in chaotic image encryption, thus providing a better guarantee for the confidentiality of image data [23,24]. Building upon this premise, this paper modifies the model and integrates it with trigonometric functions to introduce a novel 2D hyperchaotic map. This mapping can be expressed mathematically as Equation (2).

$$\begin{cases} x(i+1) = cos(asin(\pi x(i)/4) + 3 \times exp(-2x(i))sinh(by(i))) \\ y(i+1) = cos(asin(\pi y(i)/4) + 3 \times exp(-2y(i))sinh(bx(i))) \end{cases} \tag{2}$$

Here, a and b represent system parameters, with b greater than 3.5 or less than -3.5, and $x(0)$ is not equal to $y(0)$.

Through further discussion, the universal structure of this chaotic system is established as follows in Figure 1, where the function F represents the trigonometric function, the function G and the function H represent the nonlinear memristor model, such as the chaotic system function G mentioned above represents the $3 \times exp(-2x(i))sinh(by(i))$.

$$\begin{cases} x(i+1) = sin(acsc(x(i)) + 3 \times exp(-2x(i))sinh(by(i))) \\ y(i+1) = sin(acsc(y(i)) + 3 \times exp(-2y(i))sinh(bx(i))) \end{cases} \tag{3}$$

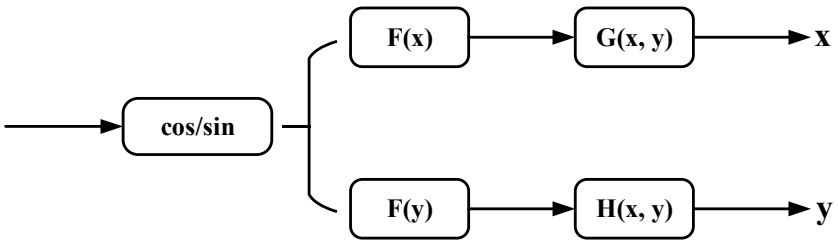
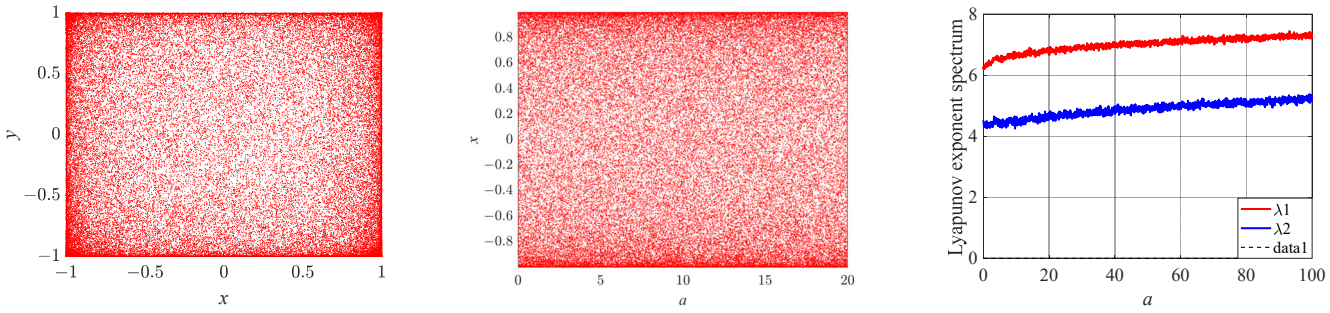


Figure 1. System framework diagram.

In order to further verify the validity of the proposed structure, a map can be mathematically represented by Equation (3). The phase diagram, bifurcation diagram, and

Lyapunov exponential spectrum of the system are briefly analyzed as shown in Figure 2, where $\lambda 1$ and $\lambda 2$ represent the two Lyapunov exponentials, and data1 represents the 0. It is shown that the system also has the optional large parameter range and complex chaotic behavior.



(**a**) Phase diagram (**b**) Bifurcation diagram (**c**) Lyapunov exponents spectrum

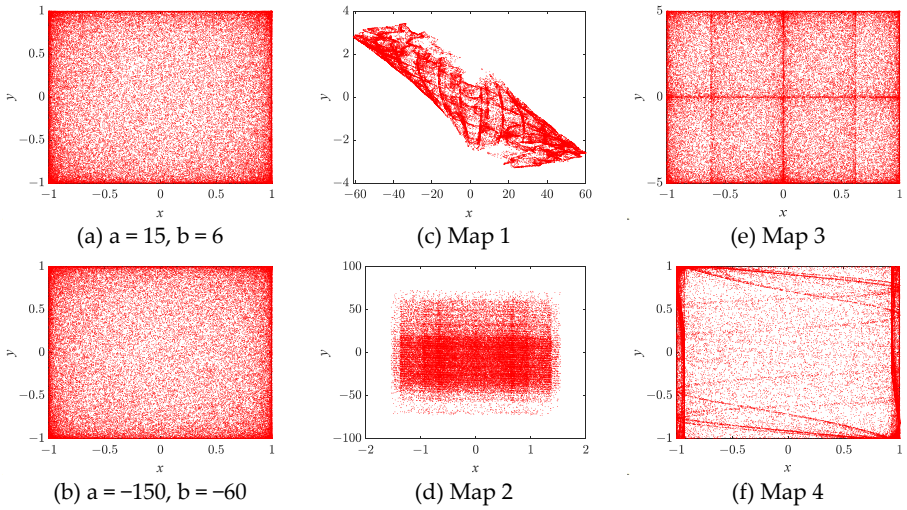
Figure 2. A new system.

3. Performance Evaluation of 2D Hyperchaotic Map

Chaotic maps are inherently nonlinear and characterized by highly complex dynamical behavior. To prove the excellent properties of the proposed 2D hyperchaotic map, this paper analyzes various aspects, including its phase diagram, bifurcation diagram, Lyapunov exponential spectrum, and 0–1 test. The proposed 2D hyperchaotic map was compared with existing systems, where Maps 1–4 correspond to the references [23–26], respectively, with their parameters drawn from the respective literature. Otherwise, Maps 1–2 is a chaotic system based on memristors.

3.1. Phase Diagram

Phase diagram analysis is one of the most straightforward techniques for examining and evaluating the performance of chaotic maps. Maps exhibiting strong chaotic behavior typically show intricate attractor trajectories that are widely and uniformly distributed within the phase diagrams. The use of phase diagrams helps to reveal the fundamental dynamic behaviors of chaotic maps and offers a preliminary insight into their complex dynamics [27]. The initial values $x_0 = 10$ and $y_0 = 2$ are set, and the proposed 2D hyperchaotic map is iterated to obtain the phase diagram of the chaotic map as shown in Figure 3a,b.



(a) a = 15, b = 6 (c) Map 1 (e) Map 3

(b) a = −150, b = −60 (d) Map 2 (f) Map 4

Figure 3. Phase diagrams for different maps.

The figure reveals that the phase diagram is evenly distributed in a large range, which can be tentatively interpreted to mean that the mapping has a complex dynamical behavior.

This paper tests the phase diagrams of Maps 1–4, as shown in Figure 3c–f, which were proposed in recent years. The comparative analysis shows that the phase diagrams of the 2D hyperchaotic map are more uniformly distributed.

3.2. Bifurcation Diagram

The bifurcation diagram can effectively and intuitively illustrate how the system's state changes with variations in parameters. As the parameters vary, the system's state undergoes corresponding adjustments [28]. Set the initial values as $x_0 = 10$ and $y_0 = 2$, and the analysis can obtain the 2D bifurcation diagram as shown in Figure 4a,b. This paper also tests the bifurcation diagrams of Maps 1–4 as shown in Figure 4c–f, which were proposed in recent years. This bifurcation diagram demonstrates that our map maintains a stable chaotic state over a wide range. Consequently, the proposed 2D hyperchaotic map exhibits characteristics of high complexity, unpredictability, and sensitivity.

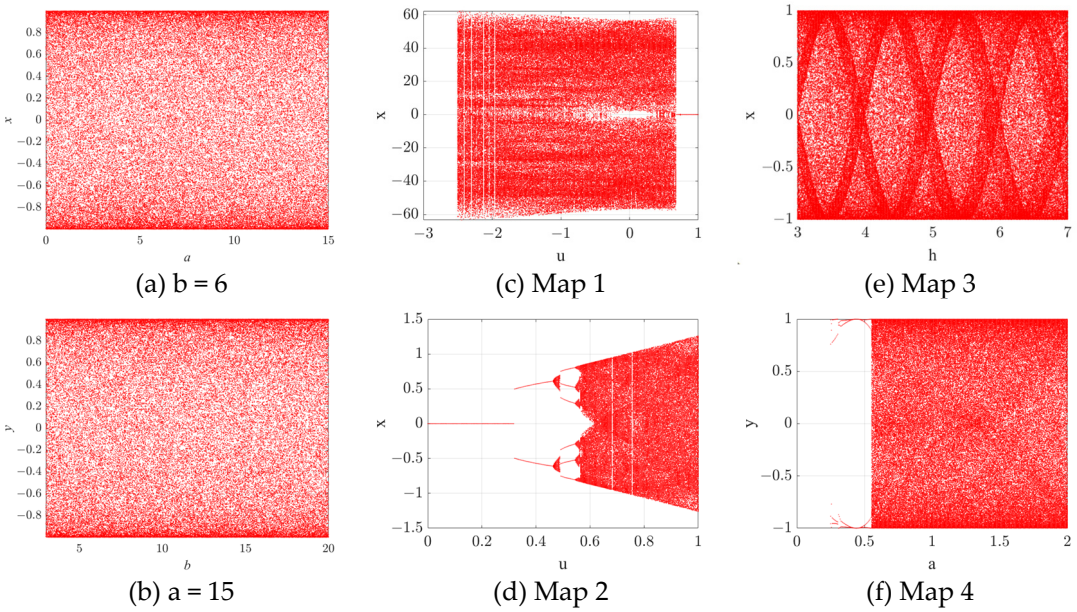
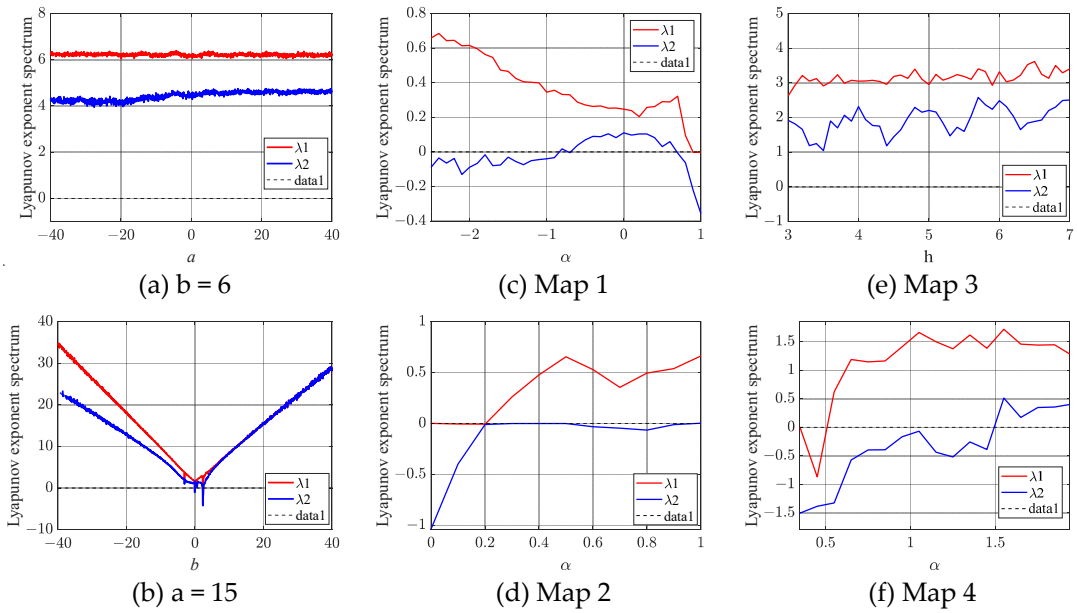


Figure 4. Bifurcation diagram.

3.3. Lyapunov Exponents Spectrum

In a 2D system with two Lyapunov exponents, the system is considered chaotic when one or both exponents are positive. Conversely, it generally remains in a periodic or quasi-periodic state [29]. In chaotic image encryption, the Lyapunov exponent is an important measure for evaluating key sensitivity, initial value sensitivity, and the extent of chaos in the system. Favorable Lyapunov exponents suggest a higher level of security and reliability for the system.

In the proposed 2D hyperchaotic map, with b held constant at 6 while a is varying within the range of [−40,40], the Lyapunov exponential spectrum is illustrated in Figure 5a. When a is fixed at 15 and b is varying within the range of [−40,40], the Lyapunov exponent spectrum is illustrated in Figure 5b. The Lyapunov exponents of Maps 1–4 as shown in Figure 5c–f are also tested, and the comparison reveals that both exponents of the system are greater than 0 and remain stable over a large range. Therefore, the system remains in a stable chaotic state, demonstrating intricate nonlinear dynamics and strong sensitivity to initial conditions.



(a) b = 6 (c) Map 1 (e) Map 3

(b) a = 15 (d) Map 2 (f) Map 4

Figure 5. Lyapunov exponents spectrum.

3.4. Complexity Analysis

PE is based on time-series data, where the system's permutation entropy is determined by partitioning the time series into distinct modes and calculating the frequency of each mode. Higher permutation entropy values reflect increased randomness and disorder in the system [30]. PE is faster and more effective in estimating complexity; thus it was chosen to analyze the complexity of the proposed 2D hyperchaotic map.

When the system parameter b is set to 6 or a is set to 15, the results are presented in Figure 6. The results demonstrate that the value of the new 2D hyperchaotic sequence surpasses 0.997, and its randomness is signifying exceptional. To further validate the sequence's randomness, this paper compares the proposed 2D hyperchaotic map and the four existing chaotic systems mentioned earlier as shown in Figure 6. The comparison highlights the higher and more stable PE complexity of the proposed 2D hyperchaotic map, indicating superior randomness.

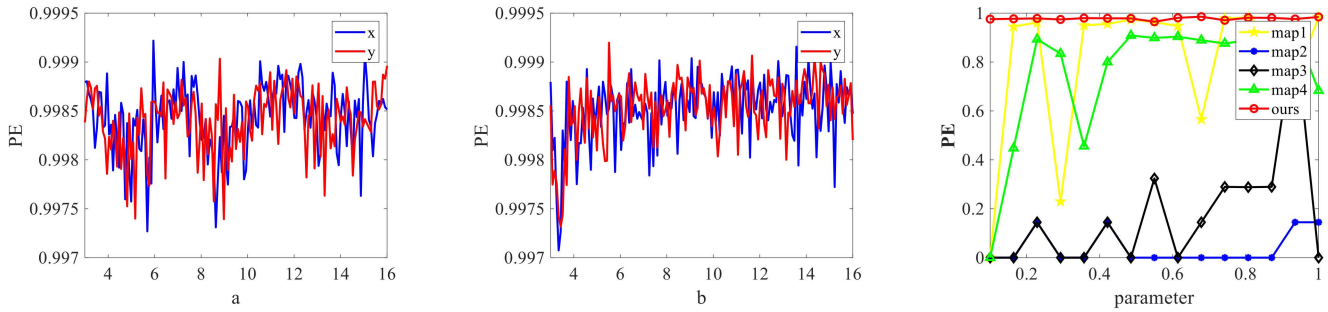


Figure 6. Permutation entropy and its comparison chart.

3.5. 0–1 Test

The 0–1 test is a method for assessing the chaotic nature of a mapping function and determining if it behaves chaotically. In contrast to the Lyapunov exponent, the 0–1 test does not require phase space reconstruction and instead focuses solely on observing changes in the input values of the map function within the interval [0,1] to

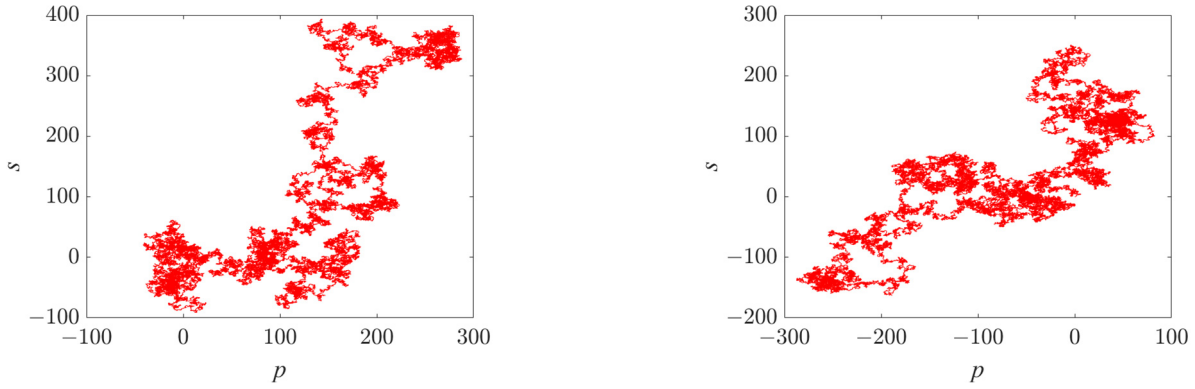
evaluate chaotic properties [31]. With a constant c, the test results of the sequence can be calculated as Equation (4).

$$K = \underset{n\to\infty}{Lim}\left(\frac{ln(M(n))}{ln(n)}\right) \tag{4}$$

where, $n = 1, 2, \cdots N$, $c \in (0, \pi)$, $M(n)$ is the displacement mean squared error, calculated as Equation (5).

$$\begin{cases} M(n) = \frac{1}{N}\sum_{i=1}^{n}\left([p(i+n) - p(i)]^2 + [s(i+n) - s(i)]^2\right) \\ p(n) = \sum_{i=1}^{n} X(i)cos(ir) \\ s(n) = \sum_{i=1}^{n} X(i)sin(ir) \end{cases} \tag{5}$$

The results of the 0–1 test are displayed in Figure 7, indicating that the trajectory follows a Brownian motion pattern and confirming the chaotic nature of the system.



(**a**) 0–1 test result of x (**b**) 0–1 test result of y

Figure 7. 0–1 test result.

4. The SEME-TDHM Algorithm

This section introduces the SEME-TDHM algorithm based on the previously discussed 2D hyperchaotic map, providing a comprehensive explanation of the encryption process. The process consists of several stages, including preprocessing, single-channel random scrambling, full-channel random scrambling, local and global diffusion, and bit-level diffusion. The decryption process, being the reverse of encryption, ensures the precise recovery of the original image.

4.1. Encryption Process

Capitalizing on the remarkable chaotic properties exhibited by the proposed chaotic map, this section introduces the SEME-TDHM algorithm grounded in the 2D hyperchaotic map. This algorithm encompasses preprocessing, scrambling, and diffusion processes, as visually depicted in Figure 8. The specific encryption procedure is detailed as follows: *A1*, *A2*, and *A3* represent the pixel values of the RGB channels, respectively.

Most encryption algorithms that use hash functions directly input image values to generate hash values, increasing the risk of the initial values being cracked. Preprocessing the image before hashing effectively mitigates this issue. Furthermore, preprocessing disrupts the pixels across the three channels at an early stage, which enhances the randomness of the image pixels obtained through subsequent encryption operations.

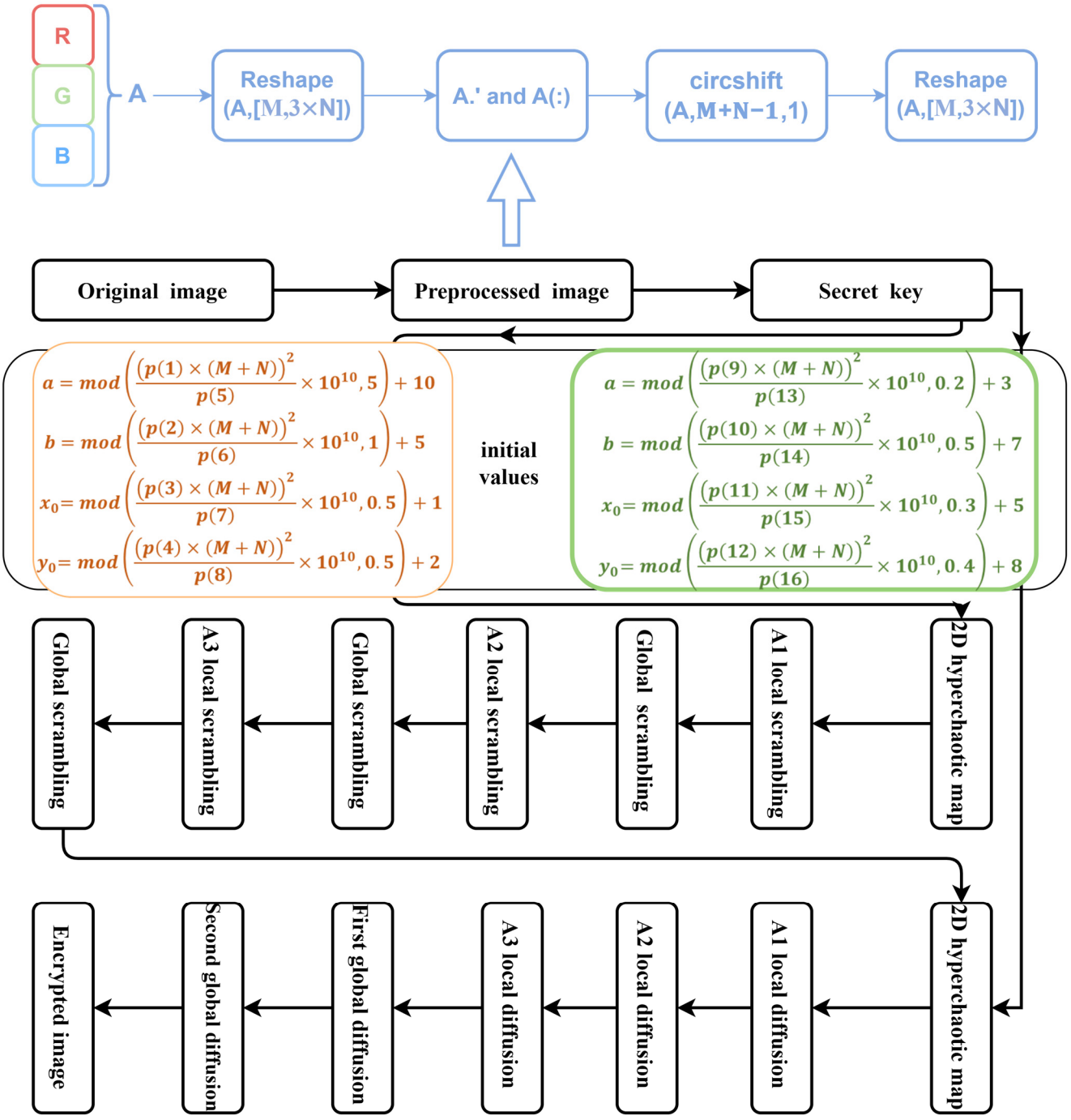


Figure 8. Diagram of the encryption process.

Step1. The R, G, and B channel pixel values are preprocessed as shown in Figure 8 and finally the result is decomposed into three $M \times N$ pixel matrices, each corresponding to the *A1*, *A2*, and *A3* of the original image;

Step2. The pixel values of the preprocessed plaintext image are used as input to the SHA−512 algorithm to generate a 128-bit string key P. According to Equation (6), P is divided into sixteen 8-bit strings, which are then converted into sixteen decimal subkeys. They will be used for the generation of initial values for scramble and diffusion, respectively, as shown in Figure 8;

$$p(i) = hex2dec(P(8 \times i - 7 : 8 \times i)), i = 1 \cdots 16 \tag{6}$$

Step3. Single-channel random scrambling: A chaotic sequence is used to generate the selection sequence *T1(i)* and the corresponding shift sequence *Q(i)*, which was used for the first pixel block *A1* to perform a row and column random shift as in Equation (7):

$$\begin{cases} A1(T1(i),:) \overset{shift\ right\ by\ Q(i)\ units}{\rightarrow} A1(T1(i),:), T1(i) < M+1, \\ A1(:,T1(i)-M) \overset{shift\ upward\ by\ Q(i)\ units}{\rightarrow} A1(:,T1(i)-M), T1(i) \geq M+1 \end{cases} \tag{7}$$

where $T1(i) \in [1, M+N]$, $Q(i) \in [0, M]$;

Step4. Full channel random scrambling: In the previous step, *P1* for $M \times 3N$ was obtained. In this step, the chaotic system is used to generate the index sequences *s1*, *s2* and the offset sequences *uu1*, *uu2*. The rows and columns of *P1* are sorted according to Algorithm 1, with the row and column index sequences randomly shifted before each sort. Figure 9 provides an example containing 5×5 images with the steps outlined below;

Algorithm 1. Global Scramble

Input: pixel matrix ***P1*** after single-channel scrambling, random sequence ***uu1*** and ***uu2***, sequence ***s1*** with index **1** to ***3N***, sequence ***s2*** with index **1** to ***M***

```
c = P1(1,:);
sorted_c = c(:,s1);
P1(1,:) = sorted_c;
for i = 2:M
    s3 = circshift(s1,uu1(i-1));
    c = P1(i,:);
    sorted_c = c(:,s3);
    P1(i,:) = sorted_c;
end
d = P1(:,1);
sorted_d = d(s2,:);
P1(:,1) = sorted_d;
for i = 2:3 × N
    s4 = circshift(s2,uu2(i − 1));
    d = P1(:,i);
    sorted_d = d(s4,:);
    P1(:,i) = sorted_d;
end
```

Output: matrix ***P1*** obtained after row global transformation

Step5. Extract the second block of pixels from *P1*. Repeat steps 3 and 4 to complete the random scrambling from single channel to full channel and then perform the same operation on the third block of pixels;

Step6. Locate diffusion matrix: The chaotic system is used to generate values from 0 to $3M \times N$ and reconstruct it into a $3M \times N$ matrix *H*. As mentioned earlier, *P1* is further divided into *A1*, *A2*, and *A3*. The matrix *H* undergoes the processing described by Equation (8) and decomposed into *A11*, *A12*, and *A13*, each of size $M \times N$, which serve as the locate diffusion matrix of *A1*, *A2*, and *A3*, respectively;

$$\begin{cases} H = 0, H < M \times N + 1 \\ H = 2, H > 2 \times M \times N \\ H = 1, else \end{cases} \tag{8}$$

Step7. Local cross-diffusion: The positions of the elements in matrix *A11* with values 1 and 2 are identified and the elements at the corresponding positions in *A1* perform XOR operation with *A2* and *A3*, respectively. As shown in Algorithm 2, *A12* then locates the positions of 0 and 2, where *indices_i* (*i* = 0, 1, 2) denotes where *i* is located. *A13* locates the position of 0 and 1;

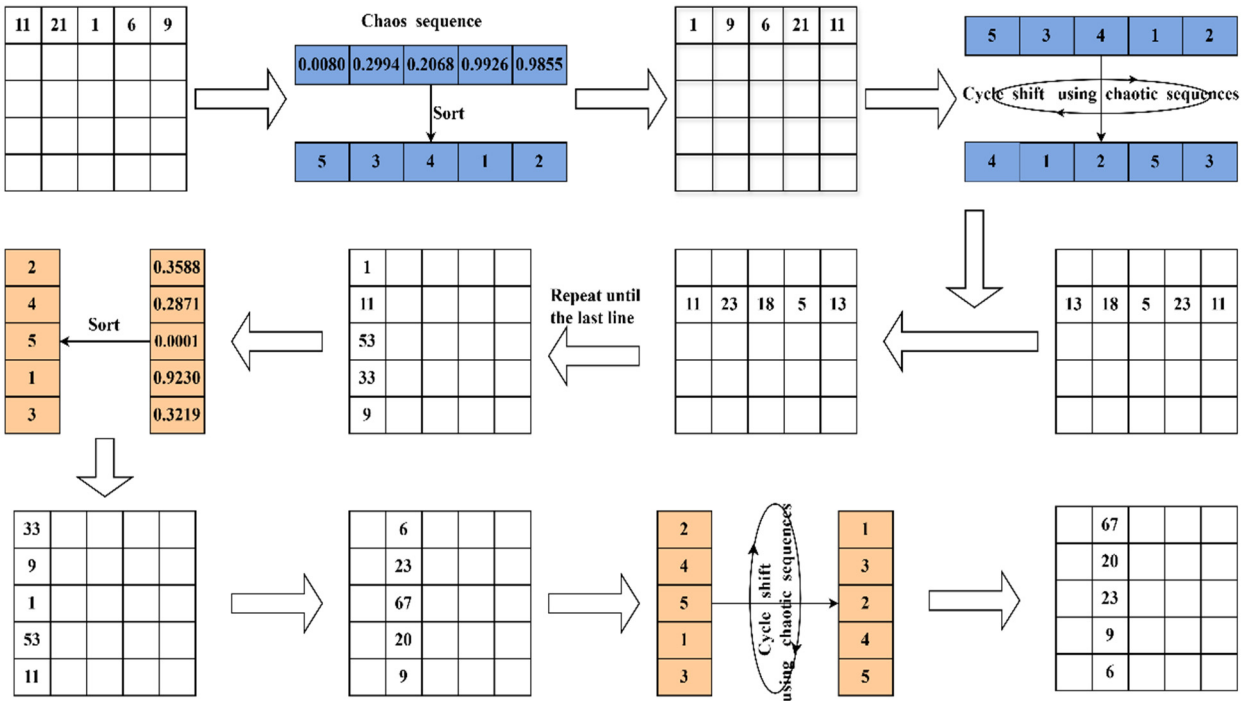


Figure 9. Global scrambling.

Algorithm 2. Locate diffusion

Input: block matrices ***A1***, ***A2***, ***A3***, Locate diffusion matrices ***A11***, ***A12***, ***A13***, all of size $M \times N$

indices_1 = find(***A11*** = =1);
A1(indices_1) = bitxor(***A1***(indices_1), ***A2***(indices_1));
indices_2 = find(***A11*** = =2);
A1(indices_2) = bitxor(***A1***(indices_2), ***A3***(indices_2));
indices_0 = find(***A12*** = = 0);
A2(indices_0) = bitxor(***A2***(indices_0), ***A1***(indices_0));
indices_2 = find(***A12*** = =2);
A2(indices_2) = bitxor(***A2***(indices_2), ***A3***(indices_2));
indices_0 = find(***A13*** = = 0);
A3(indices_0) = bitxor(***A3***(indices_0), ***A1***(indices_0));
indices_1 = find(***A13*** = =1);
A3(indices_1) = bitxor(***A3***(indices_1), ***A2***(indices_1));

Output: random matrices ***A1***, ***A2***, ***A3*** after initial diffusion

Step8. Pixel-level global diffusion: Employ the chaotic system to generate three $M \times N$ chaotic matrices, namely *x1*, *y1*, and *z1*. XOR the pixels of *A1* with those of *A2* and *x1*, XOR the pixels of *A2* with those of *A3* and *y1*. XOR the pixels of *A3* with those of *A1* and *z1*;

Step9. Bit-level global diffusion: Convert *A1*, *A2*, and *A3* into 8-bit binary sequences, then combine them into an $M \times 24N$ matrix labeled *PA*. The specific process of diffusion is shown as Equation (9):

$$\begin{cases} PA(i,:) \overset{shift\ right\ by\ i\ units}{\rightarrow} PA(i,:), i \in [1, M-1], \\ PA(i,:) \overset{shift\ right\ by\ 1\ units}{\rightarrow} PA(i,:), i = M, \\ PA(:,j) \overset{shift\ upward\ by\ j\ units}{\rightarrow} PA(:,j), j \in [1, 12 \times N], \\ PA(:,j) \overset{shift\ upward\ by\ j-12\times N\ units}{\rightarrow} PA(:,j), j \in [12 \times N + 1, 24 \times N] \end{cases} \quad (9)$$

Step10. Finally, each element is converted into a decimal number by incorporating the respective weights, and the results are stored in the *P2* matrix. These calculations complete the conversion of the encrypted image.

4.2. Decryption Process

The decryption process is the inverse of the encryption process, and its primary processes are as follows:

Step1. Bit-level global inverse diffusion: The image is transformed into a binary image matrix, which is subjected to vertical counter-diffusion and horizontal counter-diffusion in the opposite direction to the encryption process;

Step2. Pixel-level global inverse diffusion: Employ the same method as in pixel-level global diffusion but from the third-pixel block onwards, proceeding sequentially;

Step3. Local Cross counter diffusion: Local counter-diffusion aligns with the encryption process;

Step4. Full-channel random inverse scrambling: Reorder the rows of the image based on the index values generated by the chaotic sequence and cyclically shift the index sequence in an orderly manner to accomplish the inverse disordering of the rows. Columns are the same way. The process involving pseudo-random code is as Algorithm 3;

Algorithm 3. Full-channel random inverse scramble

Input: recovered block matrices ***B1***, ***B2***, ***B3***, sequence ***s2*** of index $\mathbf{1 - (3 \times N)}$, sequence ***s1*** of index $\mathbf{1 - M}$, random sequences ***uu1***, ***uu2***

```
P1 = [B1, B2, B3];
        for I = 2:3 × N
            s4 = circshift(s2,uu2(i − 1));
            temp_d = P1(:,i);
            P1(s4,i) = temp_d;
        end
            temp_d = P1(:,1);
        P1(s2,1) = temp_d;
    for i = 2:M
            s3 = circshift(s1,uu1(i − 1));
            temp_c = P1(i,:);
            P1(i, s3) = temp_c;
        end
          temp_c = P1(1,:);
        P1(1, s1) = temp_c;
```

Step5. Single-channel random inverse scrambling: After dividing the image matrix P1, apply cyclic shifting to the block pixel matrix following the division. This movement is in the opposite direction of the encryption process. Three rounds of cycling are carried out as shown in the encryption process;

Step6. Inverse preprocessing: To obtain a decrypted image, recover pixel values in an orderly fashion through unfolding, transposing, shifting, and reconstructing.

5. Experimental Results and Performance Analysis

In this section, images of different sizes have been selected to verify the effectiveness of the proposed SEME-TDHM algorithm. The chosen experimental subjects are depicted in Figure 10, where figures (a), (b), and (c) are the original images, encrypted, and decrypted images, respectively. As observed in the figures, the encrypted images do not have any information about the original image, and the decrypted image does not contain invalid noise. The employed SEME-TDHM algorithm effectively encrypts and decrypts images, yielding commendable results.

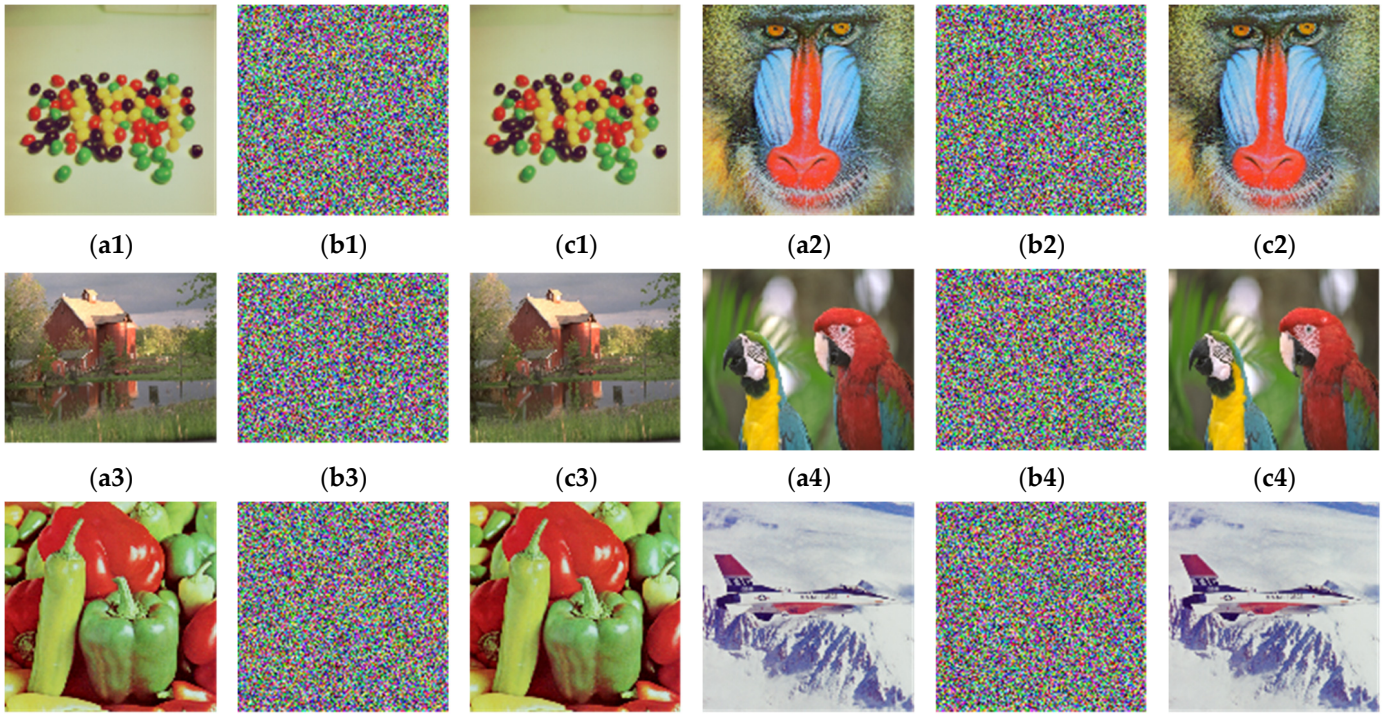
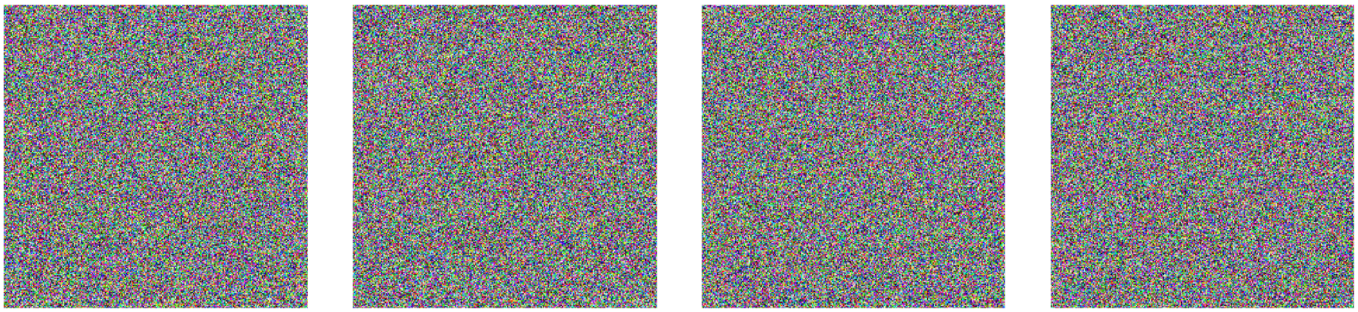


Figure 10. Encrypted and decrypted images, original images are (**a1**) Candy, 256 × 256, (**a2**) Baboon, 512 × 512, (**a3**) House, 614 × 409, (**a4**) Bird, 768 × 512, (**a5**) Pepper, 1000 × 1000, (**a6**) Aircraft, 1024 × 1024; its corresponding encrypted images are (**b1**–**b6**); its corresponding decrypted images are (**c1**–**c6**).

5.1. Key Space and Key Sensitivity Analysis

One of the performance criteria for image encryption algorithms is their ability to withstand brute-force attacks effectively, which necessitates a sufficiently large key space. In this paper, the SEME-TDHM algorithm's secret key primarily comprises the parameters a_0, b_0 of the hyperchaotic map, two discard lengths, and the initial values x_0, y_0 of the hyperchaotic map. To ensure the security and effectiveness of the encryption, the key space must be no less than 2^{128} [32,33]. Given the computational precision of 10^{-15}, the SEME-TDHM algorithm's key space is 10^{150}, which exceeds 2^{256}, significantly surpassing the minimum required key space. Consequently, the key space is sufficiently large to provide protection against brute-force attacks and potential threats from other cryptanalysis algorithms.

Meanwhile, the encrypted Candy images were decrypted using four keys slightly different from the correct key, and the decrypted results are shown in Figure 11. It is clear that these images are completely different from the correctly decrypted images. Therefore, it can be concluded that the SEME-TDHM algorithm has the ability to resist known plaintext attacks.



(**a**) $a + 10^{-15}$ (**b**) $b + 10^{-15}$ (**c**) $c + 10^{-15}$ (**d**) $d + 10^{-15}$

Figure 11. Decryption with different keys.

5.2. Information Entropy

Information entropy is a measure of the uncertainty of the average of a random variable. It evaluates the degree of diversity and unpredictability present in image data. The formula for computing information entropy is as Equation (10):

$$H(channel) = -\Sigma p(pixel)\log_2(p(pixel)) \tag{10}$$

where *p(pixel)* is the probability of a pixel value in the channel.

For a color image, the information entropy of each channel is calculated separately. These channel-specific entropies are then weighted and averaged according to their proportions in the image to obtain the overall information entropy of the color image [34]. This approach provides a thorough assessment of the image's information richness and uncertainty. An information entropy of 8 signifies the highest degree of uncertainty and maximum information.

As illustrated in Table 1, the information entropy of the SEME-TDHM algorithm is calculated and compared with that of existing algorithms. The results demonstrate that the information entropy of the encrypted images in this study approaches the ideal value of 8, surpassing other algorithms. This indicates a greater level of randomness, underscoring the robustness, security, and effectiveness of the encryption algorithm.

5.3. Histograms

Histogram counts the occurrences of each pixel value or determines the probability of each value by examining each pixel in the image. In general, the histogram of an unencrypted image displays distinct brightness and darkness levels, forming concave and convex patterns with clearly defined distributions of individual pixel values [35].

In contrast, the histogram of an encrypted image, especially one using an ideal cryptographic method, shows uniform pixel value frequencies and an even distribution throughout the image. As depicted in Figure 12, the histograms of both the original image and its corresponding encrypted image are displayed. The results unequivocally reveal that the histogram of the encrypted image exhibits a uniform distribution, effectively enhancing its ability to resist attacks.

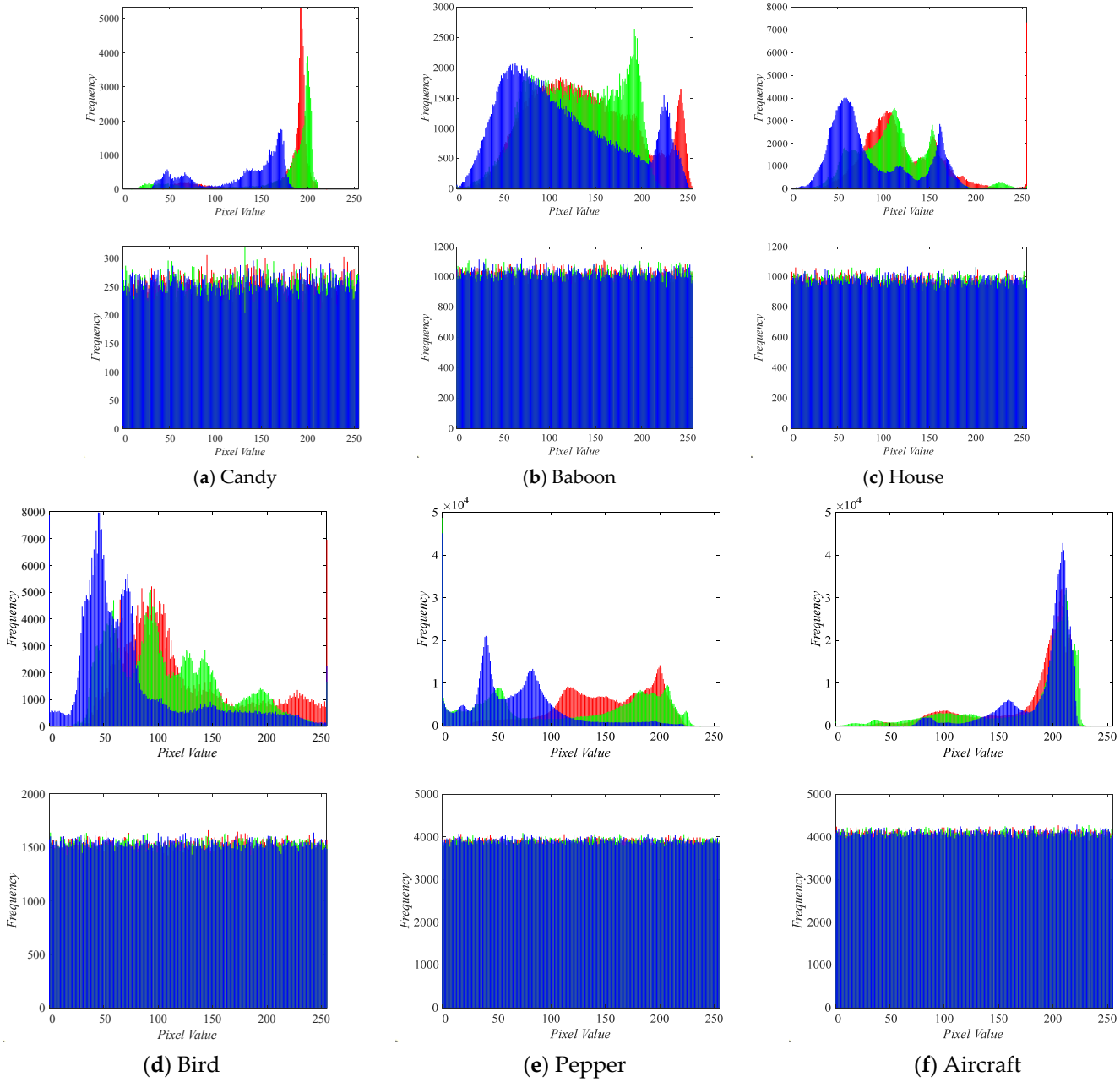


Figure 12. Histogram analysis of images and cipher images: Red represents the R channel; green represents the G channel; blue represents the B channel.

5.4. Correlation of Adjacent Pixels

A key objective of encryption algorithms is to eliminate the correlation between pixels in an image, ensuring that the encrypted image is resistant to restoration or decryption. In this paper, the correlation between pixels is effectively obscured, and the confidentiality of the encryption is strengthened by applying operations such as scrambling and diffusion to neighboring pixels [36]. These techniques significantly enhance the security of the encrypted image, making it resistant to simple restoration or decryption attempts. To

quantify the correlation among encrypted images, the correlation coefficient of pixel values is computed using Equation (11):

$$\begin{cases} E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \\ D(x) = \frac{1}{N}\sum_{i=1}^{N} [x_i - E(x)]^2 \\ Conv(x,y) = \frac{1}{N}\sum_{i=1}^{N} [x_i - E(x)][y_i - E(y)] \\ R_{xy} = \frac{Conv(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \end{cases} \tag{11}$$

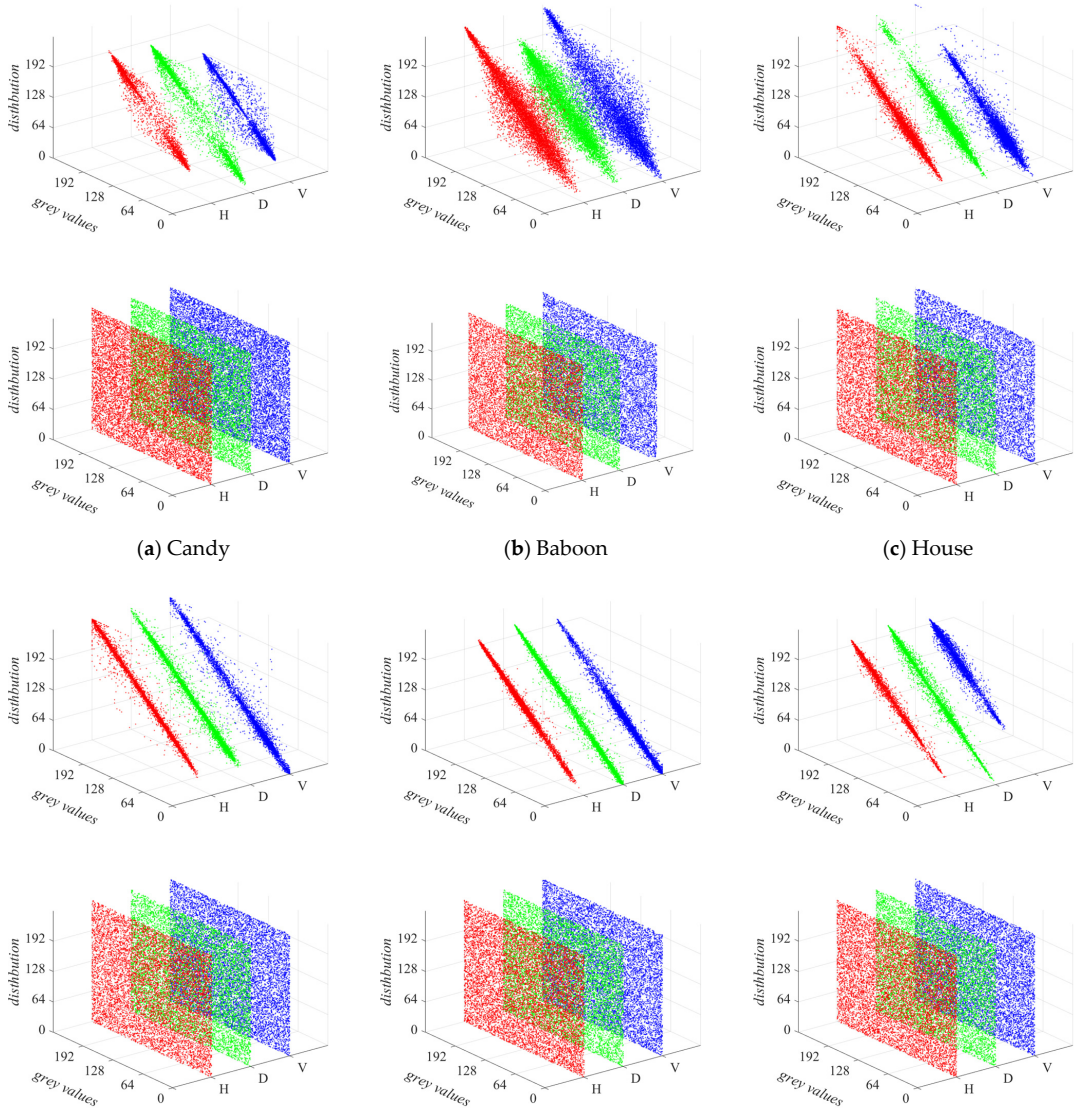
where x and y are the gray values of two neighboring pixels in the image.

Table 1. Results of information entropy.

Schemes	Images	Plain Image	R	G	B	Encrypted Image	R	G	B
Proposed	**Candy**	**6.8527**	**5.7920**	**6.2195**	**6.7986**	**7.9989**	**7.9971**	**7.9966**	**7.9976**
	Baboon	**7.7624**	**7.7067**	**7.4744**	**7.7522**	**7.9997**	**7.9993**	**7.9993**	**7.9993**
	House	**7.3791**	**7.1679**	**7.2537**	**7.1036**	**7.9998**	**7.9994**	**7.9993**	**7.9993**
	Bird	**7.6019**	**7.4699**	**7.4814**	**7.1650**	**7.9998**	**7.9995**	**7.9995**	**7.9996**
	Pepper	**7.6940**	**7.3309**	**7.5352**	**7.0630**	**7.9999**	**7.9998**	**7.9998**	**7.9998**
	Aircraft	**6.6636**	**6.7164**	**6.8005**	**6.2095**	**7.9999**	**7.9998**	**7.9998**	**7.9998**
Ref. [4]	Baboon	---	7.7067	7.4744	7.7522	---	7.9993	7.9993	7.9992
Ref. [5]	Aircraft	---	6.7178	6.7990	6.2138	---	7.9988	7.9987	7.9989
	Pepper	---	7.3006	7.5576	7.0996	---	7.9956	7.9959	7.9953
Ref. [19]	Baboon	---	---	---	---	---	7.9992	7.9993	7.9991
	Pepper	---	---	---	---	---	7.9991	7.9992	7.9992
Ref. [20]	Baboon	---	7.7470	7.4381	7.6251	---	7.9973	7.9973	7.9973
Ref. [34]	Baboon	---	---	---	---	---	7.9993	7.9994	7.9991
Ref. [35]	Baboon	---	---	---	---	---	7.9914	7.9915	7.9915
	Pepper	---	---	---	---	---	7.9911	7.9912	7.9915
Ref. [36]	Baboon	7.7742	7.7644	7.4872	7.7730	7.9989	7.9974	7.9969	7.9973
Ref. [37]	Baboon	7.5428	7.6050	7.3573	7.6662	7.9971	7.9977	7.9969	7.9969
Ref. [38]	Baboon	7.3587	7.3651	7.3975	7.3115	7.9992	7.9992	7.9993	7.9991
	Aircraft	7.5495	7.5201	7.6498	7.4787	7.9984	7.9983	7.9985	7.9985
	Pepper	7.4246	7.2018	7.6192	7.4528	7.9994	7.9996	7.9994	7.9991

Figure 13 shows the distribution of correlation coefficients among adjacent pixels in different directions. In the original image, a strong correlation is observed between neighboring pixels. Post-encryption, the pixel distribution becomes evenly spread across the image, and the correlation between neighboring pixels in various directions is negligible.

The outcomes of the correlation assessment between adjacent pixels are presented in Table 2, and obviously the pixel correlation of the encrypted image is drastically reduced. The correlation coefficient of the unencrypted image is approximated to be 1, while the correlation coefficient of the encrypted image is approximated to be 0. This demonstrates the effectiveness of the encryption process in breaking pixel correlations and enhancing the confidentiality of the image.



(**a**) Candy (**b**) Baboon (**c**) House

(**d**) Bird (**e**) Pepper (**f**) Aircraft

Figure 13. Three-channel adjacent pixel correlation of images and cipher images: red represents the distribution charts in horizontal direction; green represents the distribution charts in diagonal direction; blue represents the distribution charts in vertical direction.

Table 2. Correlation coefficients of original and encrypted images.

Images	Channels	Plaintext Image			Cipher Image		
		Diagonal	Horizontal	Vertical	Diagonal	Horizontal	Vertical
Candy	R	0.9511	0.9718	0.9758	0.0040	−0.0007	0.0048
	G	0.9469	0.9704	0.9748	−0.0039	−0.0060	0.0003
	B	0.9538	0.9736	0.9800	−0.0016	−0.0045	0.0041
Baboon	R	0.8539	0.9188	0.8702	−0.0096	0.0018	−0.0050
	G	0.7295	0.8605	0.7513	0.0061	0.0043	−0.0055
	B	0.8427	0.9113	0.8789	−0.0029	−0.0015	−0.0030
House	R	0.9479	0.9691	0.9712	−0.0029	−0.0007	−0.0044
	G	0.9292	0.9559	0.9613	−0.0055	−0.0026	−0.0009
	B	0.9467	0.9681	0.9727	−0.0020	0.0052	−0.0087
Bird	R	0.9879	0.9909	0.9904	0.0050	0.0007	0.0032
	G	0.9834	0.9890	0.9857	0.0056	0.0071	−0.0097
	B	0.9861	0.9910	0.9878	−0.0075	−0.0055	−0.0080
Pepper	R	0.9883	0.9942	0.9935	0.0053	0.0081	−0.0074
	G	0.9942	0.9972	0.9968	−0.0026	−0.0037	0.0070
	B	0.9883	0.9930	0.9946	0.0077	−0.0053	−0.0012
Aircraft	R	0.9872	0.9943	0.9932	0.0003	0.0004	−0.0038
	G	0.9845	0.9910	0.9934	0.0087	−0.0053	0.0095
	B	0.9813	0.9939	0.9866	0.0041	0.0054	−0.0021

5.5. Differential Attacks

Differential attack is a method of attacking image encryption algorithms based on small variations in pixels, by analyzing the differences between pixels in an encrypted image to break the encryption algorithm or restore the original image. Attackers use these differential characteristics to perform statistical analysis, which may lead to uncovering the encryption key or reconstructing the original image [32]. A strong encryption algorithm demonstrates extreme sensitivity to even the smallest changes in the image, meaning that altering a single pixel in the original image before encryption will result in a completely different encrypted image. In the context of this paper, the Uniform Average Change Intensity (UACI) and Number of Pixels Change Rate (NPCR) metrics are employed to test the differential attacks, with their calculations given by Equation (12) and Equation (13):

$$\begin{cases} NPCR = \frac{\sum_{ij} D(i,j)}{M \times N} \times 100\% \\ UACI = \frac{1}{M \times N} \left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255} \right] \times 100\% \end{cases} \tag{12}$$

$$D(i,j) = \begin{cases} 0, C_1(i,j) = C_2(i,j) \\ 1, C_1(i,j) \neq C_2(i,j) \end{cases} \tag{13}$$

Here, M and N represent the image's rows and columns, while C_1 and C_2 refer to the encrypted images before and after altering the pixel values in the original image, respectively. The optimal UACI and NPCR are 33.4635% and 99.6094%, respectively. In the tests, the value of individual pixels in the original image was modified to obtain the test results shown in Table 3. The performance of the SEME-TDHM algorithm was compared with other proposed algorithms. The results indicate that the UACI and NPCR values of the algorithm are very close to the ideal value and outperform those of the other algorithms. Therefore, the SEME-TDHM algorithm is highly resistant to different attacks.

Table 3. Mean values of NPCR and UACI (%).

Schemes	Images	NPCR(%)				UACI(%)			
		R	G	B	Average	R	G	B	Average
Proposed	**Candy**	**99.6162**	**99.6116**	**99.6238**	**99.6172**	**33.4038**	**33.4758**	**33.4979**	**33.4592**
	Baboon	**99.6277**	**99.5960**	**99.6227**	**99.6155**	**33.4650**	**33.4630**	**33.4543**	**33.4607**
	House	**99.6257**	**99.6117**	**99.6121**	**99.6165**	**33.4427**	**33.4959**	**33.4661**	**33.4682**
	Bird	**99.6145**	**99.6071**	**99.6269**	**99.6162**	**33.4334**	**33.4754**	**33.4828**	**33.4639**
	Pepper	**99.6082**	**99.6198**	**99.6251**	**99.6177**	**33.4833**	**33.4647**	**33.4500**	**33.4660**
	Aircraft	**99.6146**	**99.6191**	**99.6223**	**99.6187**	**33.4667**	**33.4434**	**33.4574**	**33.4558**
	Mean	**99.6178**	**99.6108**	**99.6221**	**99.6169**	**33.4658**	**33.4830**	**33.4697**	**33.4623**
Ref. [3]	Baboon	99.6300	99.6300	99.6300	99.6300	33.4200	33.4400	33.5000	33.4533
	Pepper	99.6300	99.6300	99.6300	99.6300	33.4300	33.4400	33.4300	33.4300
Ref. [4]	Baboon	99.6227	99.6304	99.6052	99.6194	33.3952	33.4674	33.4584	33.4403
	Aircraft	99.6070	99.5983	99.6006	99.6020	33.4498	33.4504	33.4792	33.4598
Ref. [19]	Baboon	99.6299	99.6037	99.6090	99.6142	33.4236	33.4326	33.4886	33.4482
	Pepper	99.5899	99.6037	99.6090	99.5997	33.4482	33.4100	33.5403	33.4661
Ref. [20]	Baboon	---	---	---	99.6053	---	---	---	33.6059
Ref. [26]	Baboon	99.5863	99.6063	99.6100	99.6008	33.4855	33.4542	33.4989	33.4795
	Pepper	99.6071	99.6044	99.6143	99.6086	33.4829	33.4417	33.4553	33.4599
Ref. [34]	Baboon	99.6010	99.6334	99.6246	99.6196	33.4455	33.4965	33.4772	33.4730
	Pepper	99.6107	99.6120	99.6066	99.6097	33.4779	33.4486	33.4785	33.4683
Ref. [35]	Baboon	99.6055	99.6164	99.6256	99.6158	33.4478	33.4495	33.4401	33.4458
Ref. [38]	Baboon	99.6664	99.6415	99.6682	99.6587	33.5075	33.5722	33.5226	33.5341
	Aircraft	99.6809	99.6043	99.6467	99.6430	33.4773	33.4223	33.4280	33.4425
Ref. [39]	Baboon	99.6174	99.6117	99.6166	99.6152	33.3923	33.3923	33.5534	33.4460
	Pepper	99.6098	99.6106	99.6322	99.6175	33.4131	33.4946	33.4067	33.4373

5.6. Noise and Data Loss Attacks

Images often experience interference and distortion during transmission, especially due to different types of noise [37]. In this study, experiments were carried out to examine the effects of gaussian noise and pepper and salt noise. Varied intensities of noise were added to the encrypted image for subsequent decryption, enabling us to assess the SEME-

TDHM algorithm's resistance to noise. The test results presented in Figure 14 demonstrate that the essential information from the original image is effectively recovered, underscoring the SEME-TDHM algorithm's robustness in the face of noise interference.

Images are susceptible to loss during transmission or storage. To comprehensively assess the SEME-TDHM algorithm's performance, this study includes a simulation of data loss by cropping the encrypted image as shown in Figure 15. The results indicate that the greater the extent of image loss, the more obscured the original image appears upon decryption. However, even with significant loss, some recognizable details of the image are still visible to the naked eye. This underscores the SEME-TDHM algorithm's efficacy in resisting data loss to a certain extent.

5.7. Analysis of Operating Speed

In addition to the encryption effect, the encryption efficiency is also an important factor in evaluating the image encryption algorithm [38]. Therefore, this section evaluates the processing speed of this algorithm and compares it with other algorithms. In this paper, the encryption and decryption time of different pixel grayscale maps was tested 20 times, and the average value was selected. The test results and comparisons are shown in Table 4. It can be seen that the encryption and decryption time is a bit long; this is because the encryption process has two rounds and involves bit level encryption. These operations enable our SEME-TDHM algorithm to effectively resist both chosen plaintext attacks and chosen ciphertext attacks. Although the encryption time is appropriately extended, its encryption performance is also improved. Although the experimental environment is different, the SEME-TDHM algorithm has better efficiency compared to similar environment algorithms. Therefore, the encryption and decryption speed of the SEME-TDHM algorithm meets the basic requirements of the application and is expected to be applied in the future.

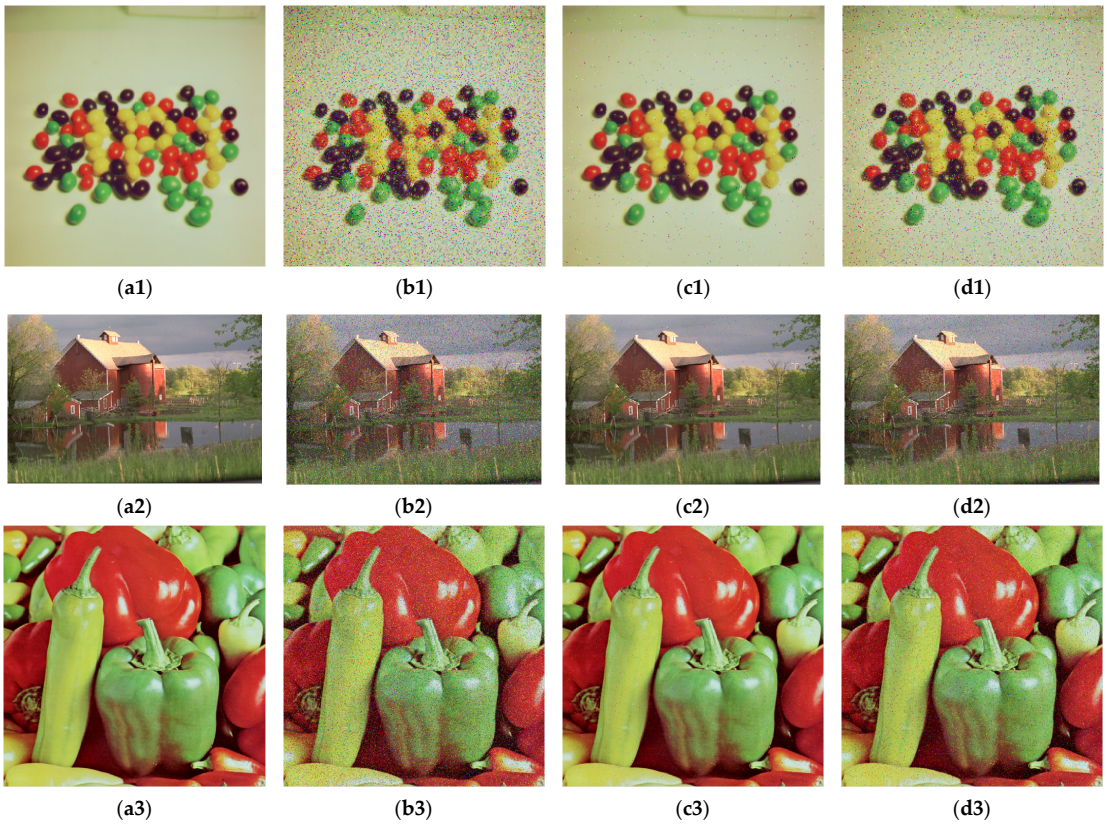


Figure 14. Noise test results of Candy, House and Pepper, Gaussian noise of (**a1**–**a3**) 1×10^{-7}, (**b1**–**b3**) 1×10^{-6}, Pepper and Salt noise of (**c1**–**c3**) 0.001, (**d1**–**d3**) 0.01.

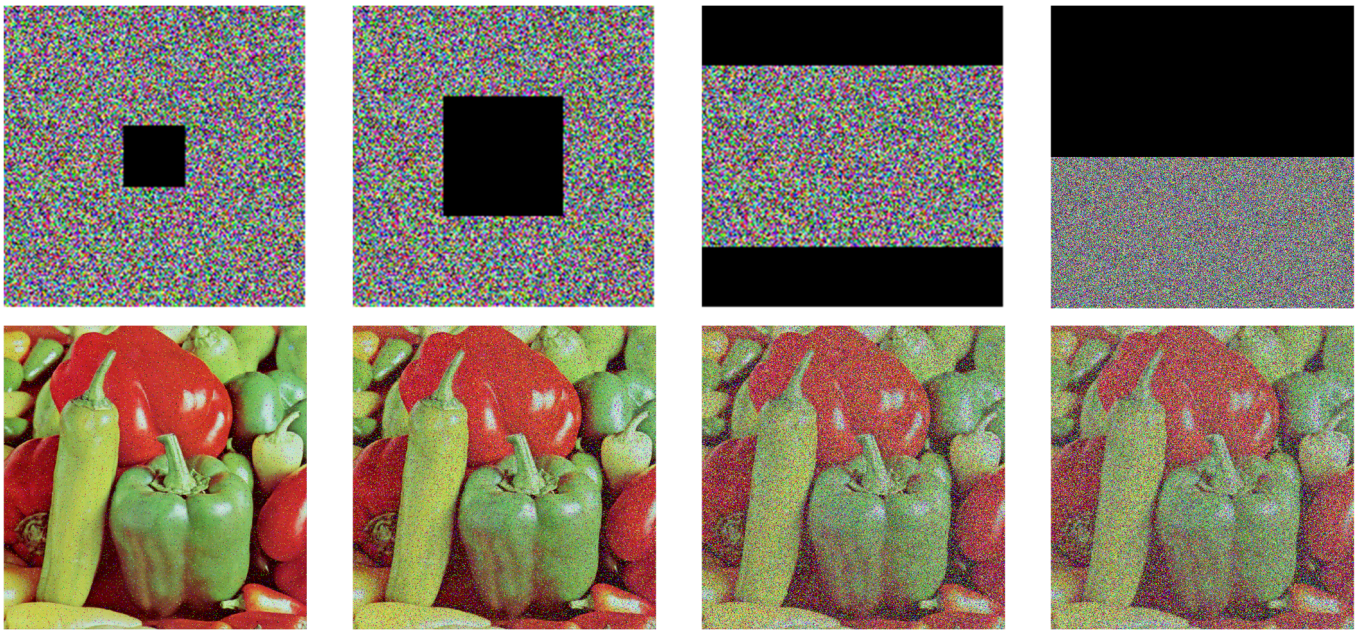


Figure 15. Data loss analysis.

Table 4. Time test of encryption and decryption (in seconds).

Image Size		Ours	Ref. [6]	Ref. [26]	Ref. [40]
256 × 256	Encryption	0.512101	2.2234	0.459837	0.699598
	Decryption	0.621802	2.2703	0.212294	0.599330
512 × 512	Encryption	1.412310	9.0016	1.769703	2.491453
	Decryption	2.401233	9.1095	0.837973	2.864108

5.8. Discussion

The SEME-TDHM encryption algorithm proposed in this paper effectively achieves image encryption and decryption. Experimental results show that the information entropy of the encrypted images remains stable around 7.999, which is better than the commonly achieved 7.99 in similar studies, indicating a higher degree of randomness. The histogram distribution of the encrypted images is uniform, with correlation coefficients remaining within three decimal places, demonstrating extremely low correlation. The UACI value of the SEME-TDHM algorithm deviates by only 0.0073 from the ideal value, while other similar studies typically show a deviation of 0.01 or more. The NPCR value deviates by only 0.0012 from the ideal value, also outperforming other studies where the deviation is usually greater than 0.01. Furthermore, the SEME-TDHM algorithm demonstrates strong resistance to Gaussian noise at 1×10^{-6} and Pepper and Salt noise at 0.01, further confirming its exceptional anti-interference capability.

6. Conclusions

This paper introduces a novel 2D hyperchaotic map based on the memristor model. Through the comprehensive analysis of the properties of the hyperchaotic system, it is proved that the system has a wide range of parameter selection and complex chaotic behavior, which is suitable for image encryption. The constructed universal structure is briefly analyzed, proving its universality and good chaotic behavior, and laying the foundation for further studies of similar systems. In addition, this paper proposes the SEME-TDHM algorithm, which uses the SHA-512 hash function and preprocessed image data for key generation. The transition from single-channel to multi-channel encryption and the introduction of chaotic indexing and positioning diffusion matrices significantly

enhance security and diffusion properties. The results show that the key space of the SEME-TDHM algorithm is sufficient, and the key sensitivity ensures that the encrypted image contains minimal statistical information, effectively resisting statistical attacks. The NPCR and UACI calculation results of the SEME-TDHM algorithm are close to the ideal values, indicating strong resistance to differential attacks. Additionally, the information entropy of the ciphertext images is very close to 8, reflecting high randomness and ideal encryption results. Although the SEME-TDHM algorithm demonstrates strong security, its practical application may face challenges related to computational efficiency and resource consumption. Future research could aim to optimize the SEME-TDHM algorithm, ensuring that it enhances encryption effectiveness while reducing computational complexity, especially when processing large-scale images. Although the SEME-TDHM algorithm is designed for color images, its core principles and structure can be adapted to secure other data types, such as video, audio, and multimedia content. By extending the SEME-TDHM algorithm to support diverse media formats, it could offer effective security solutions across a variety of applications.

Author Contributions: Conceptualization, Y.Y.; Data curation, Y.Y.; Funding acquisition, X.X.; Investigation, X.X.; Methodology, Y.Y.; Resources, X.X.; Software, Y.Y.; Supervision, Z.J.; Visualization, Z.J.; Writing—original draft, Y.Y.; Writing—review and editing, X.X. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the National Natural Science Foundation of China (Grant Nos.61927803, 61071025, 61502538, and 61501525) and the Natural Science Foundation of Hunan Province of China (Grant No. 2015JJ3157).

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

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

References

1. Liang, Z.; Chen, L.; Chen, K.; Liang, Z.; Zhu, J.; Wen, K.; Hu, Y. Holographic Encryption Algorithm Based on the New Integrated Chaotic System and Chaotic Mask. *Phys. Scr.* **2024**, *99*, 055029. [CrossRef]
2. Feng, W.; Yang, J.; Zhao, X.; Qin, Z.; Zhang, J.; Zhu, Z.; Wen, H.; Qian, K. A Novel Multi-Channel Image Encryption Algorithm Leveraging Pixel Reorganization and Hyperchaotic Maps. *Mathematics* **2024**, *12*, 3917. [CrossRef]
3. Adeel, M.; Song, Y. Secure Encryption of Color Images with Chaotic Systems and Particle Swarm Optimization. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2022**, *46*, 847–872. [CrossRef]
4. Wang, X.; Xu, X.; Sun, K.; Jiang, Z.; Li, M.; Wen, J. A Color Image Encryption and Hiding Algorithm Based On Hyperchaotic System and Discrete Cosine Transform. *Nonlinear Dyn.* **2023**, *111*, 14513–14536. [CrossRef]
5. Hu, H.; Cao, Y.; Hao, J.; Li, X.; Mou, J. A Novel Chaotic System with Hidden Attractor and Its Application in Color Image Encryption. *Multimed. Tools Appl.* **2023**, *82*, 4343–4369. [CrossRef]
6. Yan, M.; Jie, J.; Zhang, P. Chaotic Systems with Variable Indexs for Image Encryption Application. *Sci. Rep.* **2022**, *12*, 1–21. [CrossRef]
7. Li, G.; Xu, X.; Zhong, H. A Image Encryption Algorithm Based On Coexisting Multi-Attractors in a Spherical Chaotic System. *Multimed. Tools Appl.* **2022**, *81*, 32005–32031. [CrossRef]
8. Zhou, Z.; Xu, X.; Yao, Y.; Jiang, Z.; Sun, K. Novel multiple-image encryption algorithm based on a two-dimensional hyperchaotic modular model. *Chaos Solitons Fractals* **2023**, *173*, 113630. [CrossRef]
9. Dai, W.; Xu, X.; Song, X.; Li, G. Audio Encryption Algorithm Based on Chen Memristor Chaotic System. *Symmetry* **2021**, *14*, 17. [CrossRef]
10. Lei, R.; Liu, L. A Two-Dimensional Chaotic Model and Its Application in Image Encryption. *Phys. Scr.* **2024**, *99*, 075202. [CrossRef]
11. Li, H.; Hua, Z.; Bao, H.; Zhu, L.; Chen, M.; Bao, B. Two-Dimensional Memristive Hyperchaotic Maps and Application in Secure Communication. *IEEE Trans. Ind. Electron.* **2020**, *68*, 9931–9940. [CrossRef]
12. Bao, B.; Rong, K.; Li, H.; Li, K.; Hua, Z.; Zhang, X. Memristor-Coupled Logistic Hyperchaotic Map. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 2992–2996. [CrossRef]

13. Mao, N.; Tong, X.; Zhang, M.; Wang, Z. A Hyperchaotic Image Encryption Algorithm Based On LSTM Neural Network and Lifting Wavelet Transform. *Phys. Scr.* **2023**, *98*, 075215. [CrossRef]
14. Xiu, C.; Fang, J.; Ma, X. Design and Circuit Implementations of Multimemristive Hyperchaotic System. *Chaos, Solitons Fractals* **2022**, *161*, 112409. [CrossRef]
15. Zhang, J.; Guo, Y.; Xu, L.; Zhu, X.; Yang, J. Hyperchaotic Circuit Design Based On Memristor and Its Application in Image Encryption. *Microelectron. Eng.* **2022**, *265*, 111872. [CrossRef]
16. Sahin, M.E. Memristive Chaotic System-Based Hybrid Image Encryption Application with AES and RSA Algorithms. *Phys. Scr.* **2023**, *98*, 075216. [CrossRef]
17. Qian, K.; Xiao, Y.; Wei, Y.; Liu, D.; Wang, Q.; Feng, W. A Robust Memristor-Enhanced Polynomial Hyper-Chaotic Map and Its Multi-Channel Image Encryption Application. *Micromachines* **2023**, *14*, 2090. [CrossRef]
18. 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]
19. Xin, J.; Hu, H.; Zheng, J. 3D Variable-Structure Chaotic System and Its Application in Color Image Encryption with New Rubik's Cube-Like Permutation. *Nonlinear Dyn.* **2023**, *111*, 7859–7882. [CrossRef]
20. Li, Y.; Bi, L.; Li, C.; Xu, K.; Li, Y. A Henon-Like Chaotic Map and Its Application in Image Encryption Combined with Compressed Sensing. *Phys. Scr.* **2022**, *98*, 015224. [CrossRef]
21. Tang, J.; Zhang, Z.; Huang, T. Two-Dimensional Cosine–Sine Interleaved Chaotic System for Secure Communication. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *71*, 2479–2483. [CrossRef]
22. Afshari, S.; Musisi-Nkambwe, M.; Esqueda, I.S.S. Analyzing the Impact of Memristor Variability on Crossbar Implementation of Regression Algorithms With Smart Weight Update Pulsing Techniques. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 2025–2034. [CrossRef]
23. Lai, Q.; Lai, C. Design and Implementation of a New Hyperchaotic Memristive Map. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 2331–2335. [CrossRef]
24. Lai, Q.; Zhang, H. A New Image Encryption Method Based On Memristive Hyperchaos. *Opt. Laser Technol.* **2023**, *166*, 109626. [CrossRef]
25. Gao, X. Image Encryption Algorithm Based On 2D Hyperchaotic Map. *Opt. Laser Technol.* **2021**, *142*, 107252. [CrossRef]
26. Teng, L.; Wang, X.; Yang, F.; Xian, Y. Color Image Encryption Based On Cross 2D Hyperchaotic Map Using Combined Cycle Shift Scrambling and Selecting Diffusion. *Nonlinear Dyn.* **2021**, *105*, 1859–1876. [CrossRef]
27. Hou, W.; Li, S.; He, J.; Ma, Y. A Novel Image-Encryption Scheme Based on a Non-Linear Cross-Coupled Hyperchaotic System with the Dynamic Correlation of Plaintext Pixels. *Entropy* **2020**, *22*, 779. [CrossRef]
28. Hu, G.; Li, B. A Uniform Chaotic System with Extended Parameter Range for Image Encryption. *Nonlinear Dyn.* **2021**, *103*, 2819–2840. [CrossRef]
29. Tang, J.; Zhang, Z.; Chen, P.; Huang, Z.; Huang, T. A Simple Chaotic Model With Complex Chaotic Behaviors and Its Hardware Implementation. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *70*, 3676–3688. [CrossRef]
30. Zhang, T.; Zhao, Y.; Xu, X.; Wu, S.; Gu, Y. Solution and Dynamics Analysis of Fractal-Fractional Multi-Scroll Chen Chaotic System Based On Adomain Decomposition Method. *Chaos Solitons Fractals* **2023**, *178*, 114268. [CrossRef]
31. Wang, M.; Wang, X.; Wang, C.; Zhou, S.; Xia, Z.; Li, Q. Color Image Encryption Based On 2D Enhanced Hyperchaotic Logistic-sine Map and Two-Way Josephus Traversing. *Digit. Signal Process.* **2022**, *132*, 103818. [CrossRef]
32. Feng, W.; Zhang, J.; Chen, Y.; Qin, Z.; Zhang, Y.; Ahmad, M.; Woźniak, M. Exploiting Robust Quadratic Polynomial Hyperchaotic Map and Pixel Fusion Strategy for Efficient Image Encryption. *Expert Syst. Appl.* **2024**, *246*, 123190. [CrossRef]
33. Feng, W.; Zhang, J.; Qin, Z. A Secure and Efficient Image Transmission Scheme Based on Two Chaotic Maps. *Complexity* **2021**, *2021*, 1–19. [CrossRef]
34. Qiu, H.; Xu, X.; Jiang, Z.; Sun, K.; Xiao, C. A Color Image Encryption Algorithm Based On Hyperchaotic Map and Rubik's Cube Scrambling. *Nonlinear Dyn.* **2022**, *110*, 2869–2887. [CrossRef]
35. Zhang, Y.-Q.; He, Y.; Li, P.; Wang, X.-Y. A New Color Image Encryption Scheme Based On 2DNLCML System and Genetic Operations. *Opt. Lasers Eng.* **2020**, *128*, 106040. [CrossRef]
36. Haq, T.U.; Shah, T. 4D Mixed Chaotic System and Its Application to RGB Image Encryption Using Substitution-Diffusion. *J. Inf. Secur. Appl.* **2021**, *61*, 102931.
37. El-Latif, A.A.A.; Abd-El-Atty, B.; Venegas-Andraca, S.E. Controlled Alternate Quantum Walk-Based Pseudo-Random Number Generator and Its Application to Quantum Color Image Encryption. *Phys. A Stat. Mech. Its Appl.* **2020**, *547*, 123869. [CrossRef]
38. Es-Sabry, M.; El Akkad, N.; Merras, M.; Satori, K.; El-Shafai, W.; Altameem, T.; Fouda, M.M. Securing Images Using High Dimensional Chaotic Maps and DNA Encoding Techniques. *IEEE Access* **2023**, *11*, 100856–100878. [CrossRef]

39. Li, Z.; Peng, C.; Tan, W.; Li, L. A Novel Chaos-Based Color Image Encryption Scheme Using Bit-Level Permutation. *Symmetry* **2020**, *12*, 1497. [CrossRef]
40. Kang, X.; Guo, Z. A New Color Image Encryption Scheme Based On DNA Encoding and Spatiotemporal Chaotic System. *Signal Process. Image Commun.* **2020**, *80*, 115670.

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.