

Article

Color Image Encryption Based on a Novel Fourth-Direction Hyperchaotic System

Zhuoyi Lei ¹, Jiacheng Yang ², Hanshuo Qiu ¹, Xiangzi Zhang ³ and Jizhao Liu ^{1,*}

¹ School of Information Science and Engineering, Lanzhou University, No. 222 TianShui Road (South), Lanzhou 730000, China; leizhy21@lzu.edu.cn (Z.L.); qiuhs21@lzu.edu.cn (H.Q.)

² School of Information Engineering and Artificial Intelligence, Lanzhou University of Finance, No. 496 Duanjiatan, Lanzhou 730000, China; yangjch@lzufe.edu.cn

³ School of Psychology, Northwest Normal University, No. 967 Anning East Road, Lanzhou 730000, China; zhangxiangzi@nwnu.edu.cn

* Correspondence: liujz@lzu.edu.cn

Abstract: Neuromorphic computing draws inspiration from the brain to design energy-efficient hardware for information processing, enabling highly complex tasks. In neuromorphic computing, chaotic phenomena describe the nonlinear interactions and dynamic behaviors. Chaotic behavior can be utilized in neuromorphic computing to accomplish complex information processing tasks; therefore, studying chaos is crucial. Today, more and more color images are appearing online. However, the generation of numerous images has also brought about a series of security issues. Ensuring the security of images is crucial. We propose a novel fourth-direction hyperchaotic system in this paper. In comparison to low-dimensional chaotic systems, the proposed hyperchaotic system exhibits a higher degree of unpredictability and various dynamic behaviors. The dynamic behaviors include fourth-direction hyperchaos, third-direction hyperchaos, and second-direction hyperchaos. The hyperchaotic system generates chaotic sequences. These chaotic sequences are the foundation of the encryption scheme discussed in this paper. Images are altered by employing methods such as row and column scrambling as well as diffusion. These operations will alter both the pixel values and positions. The proposed encryption scheme has been analyzed through security and application scenario analyses. We perform a security analysis to evaluate the robustness and weaknesses of the encryption scheme. Moreover, we conduct an application scenario analysis to help determine the practical usability and effectiveness of the encryption scheme in real-world situations. These analyses demonstrate the efficiency of the encryption scheme.

Keywords: image encryption; chaos; hyperchaos; information security; symmetric encryption



Citation: Lei, Z.; Yang, J.; Qiu, H.; Zhang, X.; Liu, J. Color Image Encryption Based on a Novel Fourth-Direction Hyperchaotic System. *Electronics* **2024**, *13*, 2229. <https://doi.org/10.3390/electronics13122229>

Academic Editor: Manohar Das

Received: 27 March 2024

Revised: 30 April 2024

Accepted: 31 May 2024

Published: 7 June 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

The swift advancement of artificial intelligence and supercomputing technology has greatly enhanced computational efficiency [1–8]. In the increasingly advanced digital world, image encryption is becoming more and more important. The proposal of neural networks and neuromorphic computing also magnifies this issue [9–16]. Therefore, ensuring the security of images on the Internet has become critical [17]. In neuromorphic computing, chaos is involved in the interactions among neurons. This chaotic behavior can be used to perform complex information processing tasks, such as remote heart rate measurement [18]. Chaotic systems have high complexity and randomness and can produce pseudorandom sequences [19,20]. Additionally, it is suitable to utilize chaotic systems for secure image transmission. Chaos systems exhibit a significant level of complexity; minor adjustments to the parameters in the system have the potential to result in significant changes in the behavior of system [21,22].

In recent years, there has been a proliferation of encryption schemes put forth in academic papers. Progress in this area has primarily followed two main paths. The first path

involves the development of innovative chaotic systems and their application in encryption schemes [23]. For example, Liu et al. introduced a novel fourth-order chaotic system to enhance the security of medical images [24]. Similarly, Zhu et al. conducted a study on the utilization of a composite chaotic system that integrates sinusoidal and polynomial functions for the encryption of images [25]. Gao et al. proposed an efficient encryption method utilizing single-channel encryption and chaotic systems [26]. Moreover, Zhu et al. presented a set of multi-cavity hyperchaotic maps in m dimensions [27]. Jin et al. also contributed to this field by introducing a distinctive complex system with a hyperchaotic fractional order and investigating its synchronization properties [28]. Promising results have been obtained recently in various studies of neural networks and memristor-based hyperchaotic systems [29–33].

A second manner of proposing new encryption schemes is from the perspective of practical applications. For example, Lin et al. introduced a rapid image encryption method designed for embedded systems by incorporating mixed-sequence systems and the decorrelation operation [34]. Zhu et al. proposed a three-dimensional bit-level image encryption scheme utilizing the Rubik's cube method [17]. Kamal et al. introduced a novel image encryption method for grayscale and color medical images by employing an innovative image splitting approach focused on image blocks [35]. Wang et al. employed hash tables, Hilbert curves, and hyperchaotic synchronization to encrypt color images [36].

However, using low-dimensional chaotic systems to encrypt images is insecure [23,26]. The limited key space inherent in low-dimensional chaotic systems makes them vulnerable to brute force attacks.

In order to address these issues, high-dimensional hyperchaotic systems can be utilized instead of hyperchaotic systems with low dimensions. Hyperchaotic systems have at least two positive Lyapunov exponents. Positive Lyapunov exponents suggest that the system is significantly more sensitive and complex in various directions, exhibiting exponential growth in complexity [37–39]. Increasing the dimension of the system and adding more nonlinear terms will make the dynamics more complex. High-dimensional hyperchaotic systems are typically seen as better for encrypting images than low-dimensional chaotic systems [40].

In this work, a tenth-order hyperchaotic system is proposed based on n th-order ordinary differential equations proposed by Liu et al. [41]. The dynamic behavior of the system is highly complex, including fourth-direction hyperchaos, third-direction hyperchaos, and second-direction hyperchaos. Using this hyperchaotic system for image encryption can address the issues encountered in low-dimensional chaotic systems. Having a larger key space enhances the ability of encryption scheme to withstand brute force attacks. Due to high level of randomness in hyperchaotic systems, the distribution of pixels in encrypted images also shows a high level of randomness. As a result, the majority of the initial data can still be retrieved from the encrypted image, even after undergoing cropping attacks.

The major contributions of this work are outlined below:

- This paper presents an innovative encryption scheme that employs a fourth-direction (4D) hyperchaotic system. The encryption process involves the utilization of row and column scrambling as well as diffusion.
- A novel 4D hyperchaotic system is presented in this paper. Through dynamic analysis, it is found to exhibit highly complex dynamic characteristics. These dynamic characteristics include fourth-direction hyperchaos, third-direction hyperchaos, and second-direction hyperchaos.
- The encryption scheme under consideration has been evaluated and compared with other encryption methods based on criteria such as information entropy, histogram analysis, key sensitivity, pixel correlation, and key space. It has been confirmed in its ability to resist both cropping attacks and differential attacks, indicating its robust security characteristics. Moreover, a brief examination of the potential application scenarios for the proposed encryption scheme has been conducted.

The structure of this paper is as follows: Section 2 introduces encryption and decryption methods; Section 3 discusses a new 4D hyperchaotic system and its behavior; Section 4 delves into encryption and decryption processes by using the characteristics of the 4D hyperchaotic system and its dynamics; and Section 5 presents the experimental results and a thorough security analysis. Lastly, Section 6 provides a summary of the research presented in this paper.

2. Overview of Encryption and Decryption Schemes

The encryption scheme comprises two encryption iterations, with the decryption process being the opposite of the encryption process. Four chaotic sequences are generated through the hyperchaotic system described in Section 3, and these sequences are employed in both the encryption and decryption processes.

The chaotic sequence denoted as c_1 is applied for both row and column scrambling, while the chaotic sequence c_2 is employed for diffusion in the first encryption round. During the second encryption round, the chaotic sequence c_3 is utilized for diffusion, whereas the chaotic sequence c_4 is utilized for both column and row scrambling. In the row scrambling process, the image pixels are transformed into a linear sequence based on row-major order and subsequently rearranged in accordance with the sequence provided by the chaotic sequence. Similarly, in column scrambling, the image pixels are converted into a linear sequence following column-major order and then rearranged according to the chaotic sequence. Additionally, the diffusion operation involves executing an XOR operation between the chaotic sequence and the pixel value of the image.

The decryption process consists of two decryption rounds and is the complete opposite of the encryption process.

3. A Novel 4D Hyperchaotic System and Its Dynamic Behavior

3.1. Equations Describing a Novel 4D Hyperchaotic System

Reference [41] proposed an n th-order chaotic system. Equation (1) represents the chaotic system when $n = 10$.

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 - x_1 \\ \dot{x}_2 = x_3 - x_2 \\ \dot{x}_3 = x_4 - x_3 \\ \dot{x}_4 = x_5 - x_4 \\ \dot{x}_5 = x_6 - x_5 \\ \dot{x}_6 = x_7 - x_6 \\ \dot{x}_7 = x_8 - x_7 \\ \dot{x}_8 = x_9 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = -x_{10} - \rho \left(e^{\frac{x_9}{\phi}} \right) + \epsilon \left(e^{-\frac{x_9}{\phi}} \right) - 10 \cdot (x_8 + x_7 + x_6 \\ \quad + x_5 + x_4 + x_3 + x_2 - \frac{1}{200} \cdot x_1) \end{array} \right. \quad (1)$$

To derive a novel 4D hyperchaotic system, modifications need to be made based on Equation (1). Some nonlinear terms are coupled into Equation (1), increasing the complexity of the system. Through adjusting parameters and analyzing the dynamics of the system, a novel 4D hyperchaotic system is proposed.

The novel 4D hyperchaotic system is described as follows:

$$\begin{cases} \dot{x}_1 = x_2 \cdot x_7 - m_1 \cdot x_1 \cdot x_2 \\ \dot{x}_2 = m_2 \cdot x_3 \cdot x_4 - m_3 \cdot x_2^2 \\ \dot{x}_3 = m_4 \cdot x_4^2 - m_5 \cdot x_3 \cdot x_4 \\ \dot{x}_4 = m_6 \cdot x_5 \cdot x_2 - m_7 \cdot \text{sign}(x_4) \\ \dot{x}_5 = m_8 \cdot x_6 \cdot x_7 - x_5 \\ \dot{x}_6 = m_9 \cdot \text{sign}(x_7) - m_{10} \cdot x_6 \\ \dot{x}_7 = x_8 - m_{11} \cdot x_7 \\ \dot{x}_8 = m_{12} \cdot x_9 \\ \dot{x}_9 = m_{13} \cdot x_{10} \\ \dot{x}_{10} = -(c) \cdot x_{10} - \rho \left(e^{\frac{x_9}{\phi}} \right) + \epsilon \left(e^{-\frac{x_9}{\phi}} \right) - m_{14}(\cdot x_8 + \cdot x_7 + \cdot x_6 \\ + \cdot x_5 + \cdot x_4 + \cdot x_3 + \cdot x_2) - m_{15} \cdot x_1 \end{cases} \quad (2)$$

where $c \in [0.2, 2]$, c , ρ , and ϕ are control parameters, while the remaining parameters are regarded as constant parameters. These parameters determine the dynamic behaviors of the hyperchaotic system, and setting the initial values of the variables is essential.

When $(m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}) = (0.301, 0.3, 0.3, 1.4, 0.4, 1.8, 0.88, 1.1, 1.2, 0.98, 0.6, 1.8, 1.2, 10, 0.05)$, the hyperchaotic system is described as follows:

$$\begin{cases} \dot{x}_1 = x_2 \cdot x_7 - 0.301 \cdot x_1 \cdot x_2 \\ \dot{x}_2 = 0.3 \cdot x_3 \cdot x_4 - 0.3 \cdot x_2^2 \\ \dot{x}_3 = 1.4 \cdot x_4^2 - 0.4 \cdot x_3 \cdot x_4 \\ \dot{x}_4 = 1.8 \cdot x_5 \cdot x_2 - 0.88 \cdot \text{sign}(x_4) \\ \dot{x}_5 = 1.1 \cdot x_6 \cdot x_7 - x_5 \\ \dot{x}_6 = 1.2 \cdot \text{sign}(x_7) - 0.98 \cdot x_6 \\ \dot{x}_7 = x_8 - 0.6 \cdot x_7 \\ \dot{x}_8 = 1.8 \cdot x_9 \\ \dot{x}_9 = 1.2 \cdot x_{10} \\ \dot{x}_{10} = -(c) \cdot x_{10} - \rho \left(e^{\frac{x_9}{\phi}} \right) + \epsilon \left(e^{-\frac{x_9}{\phi}} \right) - 10(\cdot x_8 + \cdot x_7 + \cdot x_6 \\ + \cdot x_5 + \cdot x_4 + \cdot x_3 + \cdot x_2) - 0.05 \cdot x_1 \end{cases} \quad (3)$$

The control parameters are $\rho = 6 \times 10^{-7}$, $\phi = 0.026$, $\epsilon = 6 \times 10^{-7}$, $c = 1$, and the initial values of the variables are $(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$.

When $c = 1$, the Lyapunov exponents $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9, \lambda_{10}) = (0.25, 0.24, 0.19, 0.09, 0, -0.14, -0.76, -0.93, -0.98, -1.21)$, indicating hyperchaotic behavior [42]. Figures 1–3 show the hyperchaotic attractor. These attractors are observed when the variables all have initial values of 0.1.

3.2. Dynamic Behavior of the 4D Hyperchaotic System

The sign of the Lyapunov exponent is essential for classifying the behavior of systems. It is a unique tool to distinguish chaotic from hyperchaotic and quantify local dynamic stability [43]. Analyzing Lyapunov exponents can provide us with profound insights into the behavior of the hyperchaotic system.

The Kaplan–Yorke dimension and Lyapunov exponent spectrum are depicted in Figure 4 for values of c within the interval $[0.2, 2]$. In Figure 4, ten lines of different colors represent ten different Lyapunov exponents. The largest Lyapunov exponent is depicted by the red line, with the subsequent largest exponent shown by the green line. The third

largest exponent is indicated by the dark blue line, and the fourth largest exponent is represented by the mauve line. When the first four Lyapunov exponents exhibit positive values, the system demonstrates characteristics of a hyperchaotic attractor. For values of parameter c in the range 0.2 to 1.63, four Lyapunov exponents are positive. In contrast, for values of c ranging from 1.63 to 2, three Lyapunov exponents are positive.

To evaluate the complexity of hyperchaotic systems and characterize the dimension of their attractors, it is essential to calculate the Kaplan–Yorke dimension. The Kaplan–Yorke dimension can be calculated through the utilization of a specific mathematical formula:

$$D_{KY} = D + \frac{\sum_{i=1}^D LE_i}{|LE_D|} \quad (4)$$

In Equation (4), the term D_{KY} represents the Kaplan–Yorke dimension. The $\sum_{i=1}^D LE_i$ denotes the sum of Lyapunov exponents from 1 to D , and D is less than N (the number of Lyapunov exponents). There exists a maximum integer D for which $\sum_{i=1}^D LE_i$ is positive, and there exists an integer $D + 1$ for which $\sum_{i=1}^{D+1} LE_i$ is negative, and LE_D represents the D -th Lyapunov exponent.

When the parameter c falls within the interval of $[0.2, 2]$, the estimated range for the Kaplan–Yorke dimension is approximately between 4.62 and 7.45.

In a chaotic system, the equilibrium point refers to a specific stable state. Equilibrium points play an important role in enhancing our comprehension of the dynamics exhibited by chaotic systems.

Let $\dot{x}_1 = \dot{x}_2 = \dot{x}_3 = \dot{x}_4 = \dot{x}_5 = \dot{x}_6 = \dot{x}_7 = \dot{x}_8 = \dot{x}_9 = \dot{x}_{10} = 0$ in Equation (3), that is:

$$\begin{cases} 0 = x_2 \cdot x_7 - 0.301 \cdot x_1 \cdot x_2 \\ 0 = 0.3 \cdot x_3 \cdot x_4 - 0.3 \cdot x_2^2 \\ 0 = 1.4 \cdot x_4^2 - 0.4 \cdot x_3 \cdot x_4 \\ 0 = 1.8 \cdot x_5 \cdot x_2 - 0.88 \cdot \text{sign}(x_4) \\ 0 = 1.1 \cdot x_6 \cdot x_7 - x_5 \\ 0 = 1.2 \cdot \text{sign}(x_7) - 0.98 \cdot x_6 \\ 0 = x_8 - 0.6 \cdot x_7 \\ 0 = 1.8 \cdot x_9 \\ 0 = 1.2 \cdot x_{10} \\ 0 = -(c) \cdot x_{10} - \rho \left(e^{\frac{x_9}{\phi}} \right) + \epsilon \left(e^{-\frac{x_9}{\phi}} \right) - 10(\cdot x_8 + \cdot x_7 + \cdot x_6 \\ \quad + \cdot x_5 + \cdot x_4 + \cdot x_3 + \cdot x_2) - 0.05 \cdot x_1 \end{cases} \quad (5)$$

When the values of ρ , ϕ , ϵ , and c are specified as 6×10^{-7} , 0.026, 6×10^{-7} , and 1, respectively, the equilibrium point $(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ is identified. To assess the stability of the equilibrium point, it is necessary to acquire the relevant Jacobian matrix. The Jacobian matrix is a matrix consisting of the first-order partial derivatives of a function with multiple variables. The Jacobian matrix is expressed as $f'(x)$. Through the Jacobian matrix, ten eigenvalues can be calculated.

$$f'(x) = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_{10}} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_{10}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{10}}{\partial x_1} & \frac{\partial f_{10}}{\partial x_2} & \dots & \frac{\partial f_{10}}{\partial x_{10}} \end{bmatrix} \quad (6)$$

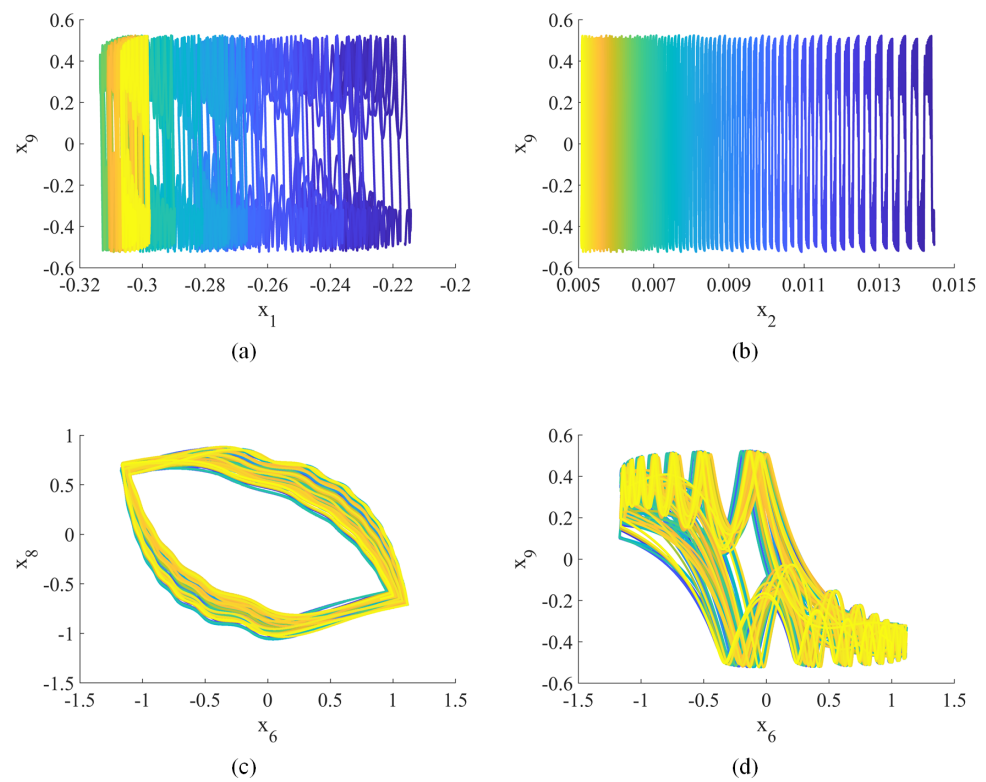


Figure 1. Hyperchaotic attractor (2D phase plane): (a) x_1 – x_9 ; (b) x_2 – x_9 ; (c) x_6 – x_8 ; (d) x_6 – x_9 .

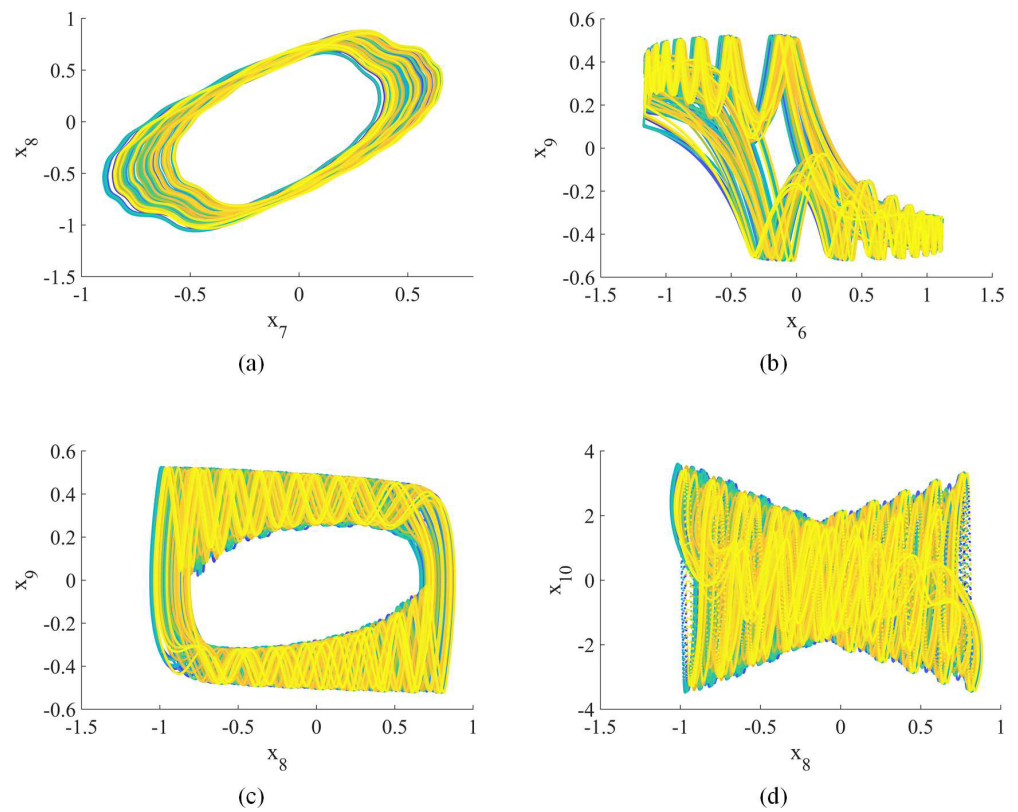


Figure 2. Hyperchaotic attractor (2D phase plane): (a) x_7 – x_8 ; (b) x_6 – x_9 ; (c) x_8 – x_9 ; (d) x_8 – x_{10} .

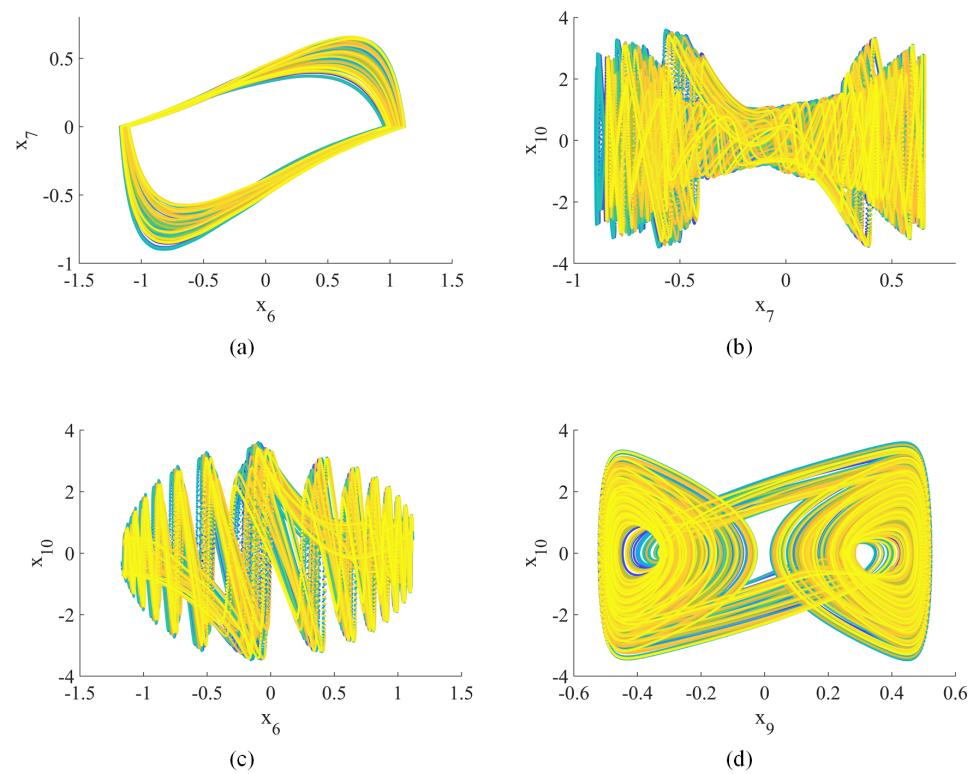


Figure 3. Hyperchaotic attractor (2D phase plane): (a) x_6 – x_7 ; (b) x_7 – x_{10} ; (c) x_6 – x_{10} ; (d) x_9 – x_{10} .

In dynamical systems, eigenvalues describe the stability of the system near its equilibrium point. The ten eigenvalues are as follows:

$$\begin{aligned}
 \lambda_1 &= -0.00030913, \\
 \lambda_2 &= (-0.00009904 + 0.00028896i), \\
 \lambda_3 &= (-0.00009904 - 0.00028896i), \\
 \lambda_4 &= (0.00024071 + 0.00017865i), \\
 \lambda_5 &= (0.00024071 - 0.00017865i), \\
 \lambda_6 &= 0, \\
 \lambda_7 &= 0, \\
 \lambda_8 &= 0, \\
 \lambda_9 &= -8.8, \\
 \lambda_{10} &= -0.00001.
 \end{aligned} \tag{7}$$

Corresponding to the eigenvalues of λ_2 and λ_3 , λ_4 and λ_5 exhibit a complex conjugate relationship. The real parts of λ_4 and λ_5 are positive, while the real parts of λ_1 , λ_2 , λ_3 , λ_9 , and λ_{10} are negative.

In a dynamic system, the presence of a positive eigenvalue indicates instability in the corresponding direction, potentially leading to instability within the system. A negative eigenvalue shows stability in that direction and helps the system maintain equilibrium. If the eigenvalue is zero, the system may have one or more degrees of freedom without a definite direction of stability.

Among the ten eigenvalues analyzed, it is noteworthy that two of them have positive real parts, indicating that the system will move away from the equilibrium point rather than return to it. Additionally, five eigenvalues have negative real parts. This shows that a slight disturbance in the system near the equilibrium point will lead to a gradual decrease in the system state in that direction, ultimately returning to the equilibrium point. Additionally,

there are three eigenvalues that are equal to zero. The state change in this direction cannot be determined. The linear change is relatively slow, making it impossible to ascertain whether it is stable or unstable.

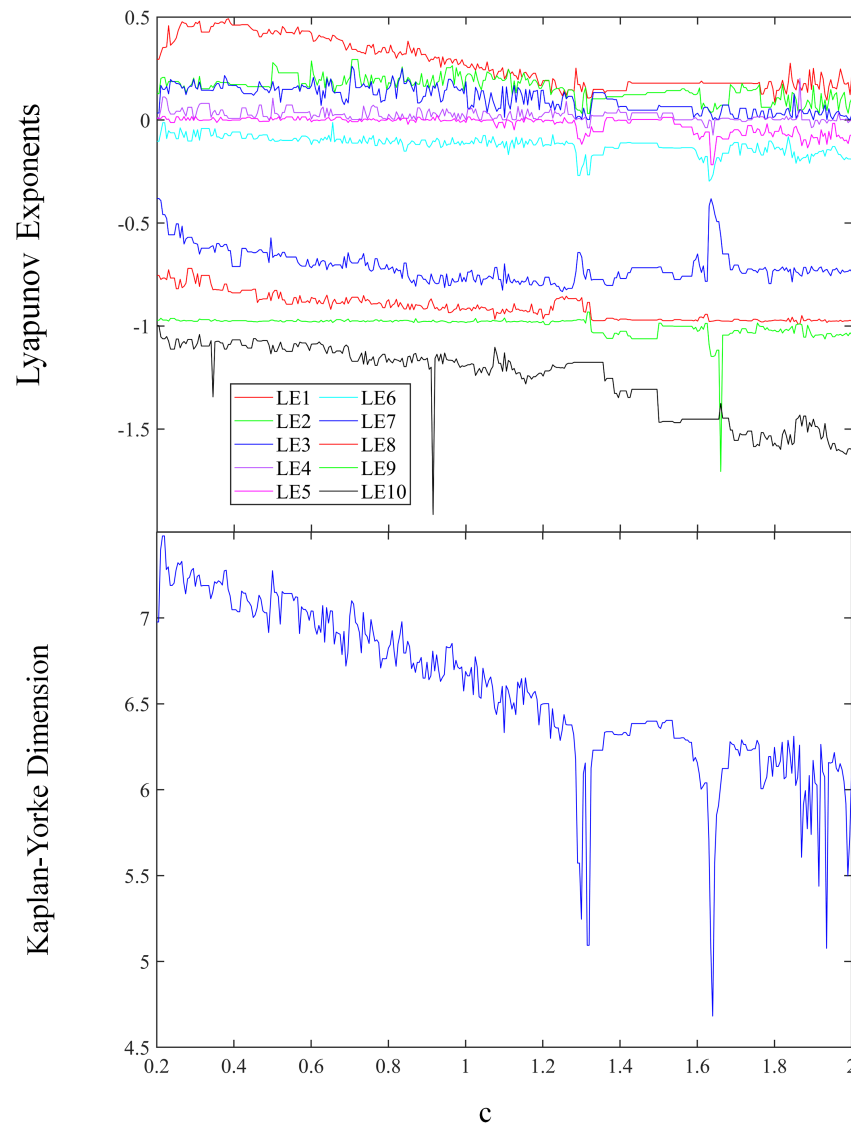


Figure 4. The Lyapunov exponent spectrum and Kaplan–Yorke dimension.

Divergence describes whether a vector field is “convergent” or “divergent” at a specific point. If the divergence of a vector field at a certain point is positive, it shows that the vectors around this specific point mainly point towards this point. In other words, more fluid or matter converges toward the certain point, suggesting that the point is a “convergence point”. In other words, more fluid or matter converges toward that point, suggesting that the point is a “convergence point”. On the contrary, if the divergence of a vector field at a certain point is negative, it shows that the vectors around the point mainly point away from it. If the divergence is zero, it means that the fluid or material around the point does not converge or diverge; in other words, the point is a “stable point”. In this system, the divergence equation is as follows:

$$\nabla \cdot \mathbf{F} = \sum_{i=1}^{10} \frac{\partial \dot{x}_i}{\partial x_i} \quad (8)$$

According to Equation (8), the calculated result is -8.8 . In general, the hyperchaotic system is commonly found to exhibit a negative divergence, indicating its inherent divergent characteristics [44].

4. Detailed Encryption and Decryption Schemes

Four pseudorandom sequences generated by the innovative 4D hyperchaotic system are used for encrypting images. The chaotic sequence denoted as c_1 is applied for both row and column scrambling, whereas the sequence c_2 is employed for diffusion during the first encryption round. In the second encryption round, the sequence c_3 is utilized for diffusion, while c_4 is employed for scrambling both rows and columns. The encryption and decryption processes are delineated in Algorithms 1 and 2, correspondingly. The generation processes of c_1 , c_2 , c_3 , and c_4 are expounded in Algorithms 3, 4, 5 and 6, respectively.

The definitions of the variables in Algorithms 1 and 2 are outlined as follows:

- *Org_Img*: Pixel matrix of the original image.
- *En_Img*: Pixel matrix of the encrypted image.
- *1st_row*: The result of the first row scrambling.
- *1st_column*: The result of the first column scrambling.
- *1st_diffusion*: The result of the first diffusion.
- *2nd_row*: The result of the second row scrambling.
- *2nd_column*: The result of the second column scrambling.
- *2nd_diffusion*: The result of the second diffusion.
- *1st_encryption*: The result of the first encryption.

Algorithm 1 Pseudocode of the encryption process

Input value: *Org_Img*

Output value: *En_Img*

- 1: $Average_pixel_value \leftarrow \text{mean2}(Org_Img) \times 10^{-9}$
 - 2: $y_1 = y_2 = y_3 = y_4 = y_5 = y_6 = y_7 = y_8 = y_9 = y_{10} = 0.1$
 - 3: $y_1(1) \leftarrow y_1(1) + Average_pixel_value$
 - 4: $c_1 \leftarrow \text{SEQA}(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10})$
 - 5: $c_2 \leftarrow \text{SEQB}(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10})$
 - 6: $1st_row \leftarrow \text{row_scrambling}(Org_Img, c_1)$
 - 7: $1st_column \leftarrow \text{column_scrambling}(1st_row, c_1)$
 - 8: $1st_diffusion(1st_encryption) \leftarrow \text{diffusion}(1st_column, c_2)$
 - 9: $y_1 = y_2 = y_3 = y_4 = y_5 = y_6 = y_7 = y_8 = y_9 = y_{10} = 0.1$
 - 10: $y_1(1) \leftarrow y_1(1) + Average_pixel_value$
 - 11: $c_3 \leftarrow \text{SEQC}(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10})$
 - 12: $c_4 \leftarrow \text{SEQD}(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10})$
 - 13: $2nd_diffusion \leftarrow \text{diffusion}(1st_diffusion, c_4)$
 - 14: $2nd_column \leftarrow \text{column_scrambling}(2nd_diffusion, c_3)$
 - 15: $2nd_row(En_Img) \leftarrow \text{row_scrambling}(2nd_column, c_3)$
-

Encryption Algorithm:

- <1> Determine the key by computing the mean value of pixels in the original image.
- <2> Divide the individual pixels in the original image based on three distinct color channels: red, green, and blue.
- <3> Calculate c_1 by applying mathematical operations to the chaotic sequence A following Algorithm 3.
- <4> Calculate c_2 through sequence B according to Algorithm 4.
- <5> Calculate c_3 through sequence C according to Algorithm 5.
- <6> Calculate c_4 through sequence D according to Algorithm 6.
- <7> Rearrange the three channels by applying row scrambling using sequence c_1 as described in <3>.

- <8> Rearrange the three channels by applying column scrambling using sequence c_1 as described in <3>.
- <9> Diffuse the three channels by employing the sequence denoted as c_2 as described in <4>.
- <10> Integrate the three channels to generate the first round encryption image.
- <11> Diffuse the three channels by employing the sequence denoted as c_4 as described in <6>.
- <12> Rearrange the three channels by applying column scrambling using sequence c_3 as described in <5>.
- <13> Rearrange the three channels by applying row scrambling using sequence c_3 as described in <5>.
- <14> Integrate the three channels to generate the second round encryption image.

Algorithm 2 Pseudocode of the decryption process

 Input value: En_Img

 Output value: Org_Img

```

1:  $y_1 = y_2 = y_3 = y_4 = y_5 = y_6 = y_7 = y_8 = y_9 = y_{10} = 0.1$ 
2:  $y_1(1) \leftarrow y_1(1) + Average\_pixel\_value$ 
3:  $c_3 \leftarrow SEQC(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10})$ 
4:  $c_4 \leftarrow SEQD(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10})$ 
5:  $1st\_row \leftarrow row\_scrambling(En\_Img, c_3)$ 
6:  $1st\_column \leftarrow column\_scrambling(1st\_row, c_3)$ 
7:  $1st\_diffusion(1st\_encryption) \leftarrow diffusion(1st\_column, c_4)$ 
8:  $y_1 = y_2 = y_3 = y_4 = y_5 = y_6 = y_7 = y_8 = y_9 = y_{10} = 0.1$ 
9:  $y_1(1) \leftarrow x_1(1) + Average\_pixel\_value$ 
10:  $c_1 \leftarrow SEQA(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10})$ 
11:  $c_2 \leftarrow SEQB(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10})$ 
12:  $2nd\_diffusion \leftarrow diffusion(1st\_encryption, c_2)$ 
13:  $2nd\_column \leftarrow column\_scrambling(2nd\_diffusion, c_1)$ 
14:  $2nd\_row(Org\_Img) \leftarrow row\_scrambling(2nd\_column, c_1)$ 

```

Algorithm 3 SEQA

```

1: function SEQA( $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}$ )
2:   for  $k = 1$  to  $m \times n$  do
3:     [ $dy_1, dy_2, dy_3, dy_4, dy_5, dy_6, dy_7, dy_8, dy_9, dy_{10}$ ]
        $\leftarrow Runge\text{-}Kutta(y_1(k), y_2(k), y_3(k), y_4(k), y_5(k), y_6(k), y_7(k), y_8(k), y_9(k), y_{10}(k))$ 
4:      $y_1(k+1) \leftarrow y_1(k) + dy_1$ 
5:      $y_2(k+1) \leftarrow y_2(k) + dy_2$ 
6:      $y_3(k+1) \leftarrow y_3(k) + dy_3$ 
7:      $y_4(k+1) \leftarrow y_4(k) + dy_4$ 
8:      $y_5(k+1) \leftarrow y_5(k) + dy_5$ 
9:      $y_6(k+1) \leftarrow y_6(k) + dy_6$ 
10:     $y_7(k+1) \leftarrow y_7(k) + dy_7$ 
11:     $y_8(k+1) \leftarrow y_8(k) + dy_8$ 
12:     $y_9(k+1) \leftarrow y_9(k) + dy_9$ 
13:     $y_{10}(k+1) \leftarrow y_{10}(k) + dy_{10}$ 
14:     $s_k = |y_3(k) \times 10^{12} - floor(y_3(k) \times 10^{12})|$ 
15:   end for
16:   [ $z1, c$ ] = sort( $s$ );
17:   return  $c$ 
18: end function

```

Algorithm 4 SEQB

```

1: function SEQB( $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}$ )
2:   for  $k = 1$  to  $m \times n$  do
3:     [ $dy_1, dy_2, dy_3, dy_4, dy_5, dy_6, dy_7, dy_8, dy_9, dy_{10}$ ]
        $\leftarrow$  Runge–Kutta( $y_1(k), y_2(k), y_3(k), y_4(k), y_5(k), y_6(k), y_7(k), y_8(k), y_9(k), y_{10}(k)$ )
4:      $y_1(k+1) \leftarrow y_1(k) + dy_1$ 
5:      $y_2(k+1) \leftarrow y_2(k) + dy_2$ 
6:      $y_3(k+1) \leftarrow y_3(k) + dy_3$ 
7:      $y_4(k+1) \leftarrow y_4(k) + dy_4$ 
8:      $y_5(k+1) \leftarrow y_5(k) + dy_5$ 
9:      $y_6(k+1) \leftarrow y_6(k) + dy_6$ 
10:     $y_7(k+1) \leftarrow y_7(k) + dy_7$ 
11:     $y_8(k+1) \leftarrow y_8(k) + dy_8$ 
12:     $y_9(k+1) \leftarrow y_9(k) + dy_9$ 
13:     $y_{10}(k+1) \leftarrow y_{10}(k) + dy_{10}$ 
14:     $s_k = \lfloor y_3(k) \times 10^8 - \text{round}(y_3(k) \times 10^8) \rfloor$ 
15:     $c_k = \text{mod}(y_2(k) \times 10^{12} - \lfloor y_2(k) \times 10^{12} \rfloor + s_k, 256)$ 
16:  end for
17:   $c(k) = \text{fix}(c(k));$ 
18:  return  $c$ 
19: end function

```

Algorithm 5 SEQC

```

1: function SEQC( $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}$ )
2:   for  $k = 1$  to  $m \times n \times 10$  do
3:     [ $dy_1, dy_2, dy_3, dy_4, dy_5, dy_6, dy_7, dy_8, dy_9, dy_{10}$ ]
        $\leftarrow$  Runge–Kutta( $y_1(k), y_2(k), y_3(k), y_4(k), y_5(k), y_6(k), y_7(k), y_8(k), y_9(k), y_{10}(k)$ )
4:      $y_1(k+1) \leftarrow y_1(k) + dy_1$ 
5:      $y_2(k+1) \leftarrow y_2(k) + dy_2$ 
6:      $y_3(k+1) \leftarrow y_3(k) + dy_3$ 
7:      $y_4(k+1) \leftarrow y_4(k) + dy_4$ 
8:      $y_5(k+1) \leftarrow y_5(k) + dy_5$ 
9:      $y_6(k+1) \leftarrow y_6(k) + dy_6$ 
10:     $y_7(k+1) \leftarrow y_7(k) + dy_7$ 
11:     $y_8(k+1) \leftarrow y_8(k) + dy_8$ 
12:     $y_9(k+1) \leftarrow y_9(k) + dy_9$ 
13:     $y_{10}(k+1) \leftarrow y_{10}(k) + dy_{10}$ 
14:    if  $\text{mod}(k, 10) = 0$  then
15:       $s_k = \lfloor y_8(k) \times 10^8 - \text{floor}(y_8(k) \times 10^8) \rfloor$ 
16:    end if
17:  end for
18:  [ $z1, c$ ] = sort( $s$ );
19:  return  $c$ 
20: end function

```

Decryption Algorithm:

- <1> Rearrange the three channels by applying row scrambling using sequence c_3 .
- <2> Rearrange the three channels by applying column scrambling using sequence c_3 .
- <3> Diffuse the three channels by employing the sequence denoted as c_4 .
- <4> Integrate the three channels to generate the first round encryption image.
- <5> Diffuse the three channels by employing the sequence denoted as c_2 .
- <6> Rearrange the three channels by applying column scrambling using sequence c_1 .
- <7> Rearrange the three channels by applying row scrambling using sequence c_1 .
- <8> Integrate the three channels to generate the second round encryption image.

Algorithm 6 SEQD

```

1: function SEQD( $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}$ )
2:   for  $k = 1$  to  $m \times n \times 10$  do
3:     [ $dy_1, dy_2, dy_3, dy_4, dy_5, dy_6, dy_7, dy_8, dy_9, dy_{10}$ ]
        $\leftarrow$  Runge-Kutta( $y_1(k), y_2(k), y_3(k), y_4(k), y_5(k), y_6(k), y_7(k), y_8(k), y_9(k), y_{10}(k)$ )
4:      $y_1(k+1) \leftarrow y_1(k) + dy_1$ 
5:      $y_2(k+1) \leftarrow y_2(k) + dy_2$ 
6:      $y_3(k+1) \leftarrow y_3(k) + dy_3$ 
7:      $y_4(k+1) \leftarrow y_4(k) + dy_4$ 
8:      $y_5(k+1) \leftarrow y_5(k) + dy_5$ 
9:      $y_6(k+1) \leftarrow y_6(k) + dy_6$ 
10:     $y_7(k+1) \leftarrow y_7(k) + dy_7$ 
11:     $y_8(k+1) \leftarrow y_8(k) + dy_8$ 
12:     $y_9(k+1) \leftarrow y_9(k) + dy_9$ 
13:     $y_{10}(k+1) \leftarrow y_{10}(k) + dy_{10}$ 
14:    if  $\text{mod}(k, 10) = 0$  then
15:       $s_k = \lfloor y_3(k) \times 10^8 - \text{round}(y_3(k) \times 10^8) \rfloor$ 
16:       $c_k = \text{mod}(y_3(k) \times 10^{12} - \lfloor y_3(k) \times 10^{12} \rfloor + s_k, 256)$ 
17:    end if
18:  end for
19:   $c(k) = \text{fix}(c(k));$ 
20:  return  $c$ 
21: end function

```

The encryption and decryption processes are illustrated in Figures 5 and 6.

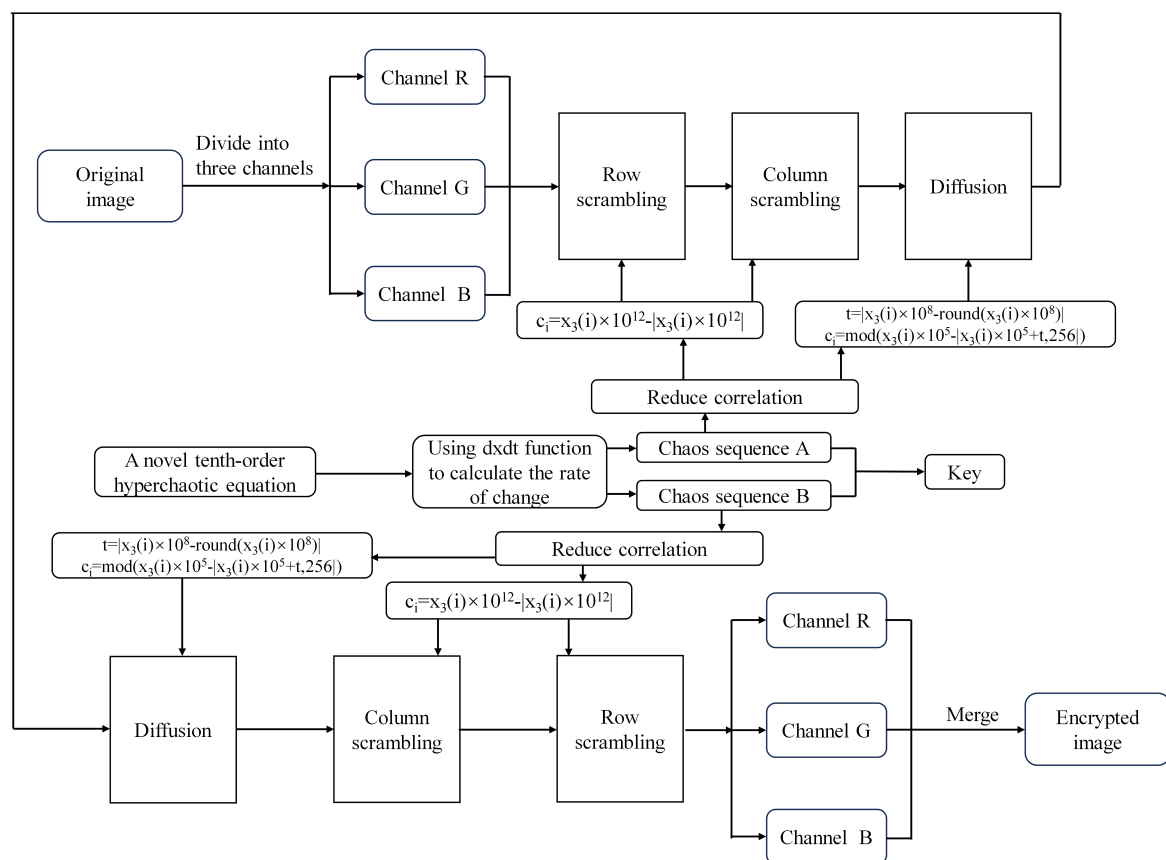


Figure 5. The process of encrypting an image.

To evaluate encrypted images, the variance of the histogram can be used as a method of measuring the histogram. A lower variance value signifies a higher level of uniformity in encrypted result. The equation for the computation of histogram variance is outlined as follows:

$$\text{var}(X) = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k (x_i - x_j)^2 \quad (9)$$

The variance of the histogram is denoted by X . The variables x_j and x_i denote the pixel values of positions j and i in a single channel. In this study, an analysis using histograms was conducted on the image titled “Jetplane”. Table 1 presents the variance values of the image across the three channels. The variance values observed in the encrypted image were significantly lower than those in the original image across all three color channels. The difference suggests a more consistent spread of pixel values in the encrypted image, effectively hiding the original image’s structure and features. Therefore, it is impossible for an attacker to extract any useful information from the encrypted image just by observation.

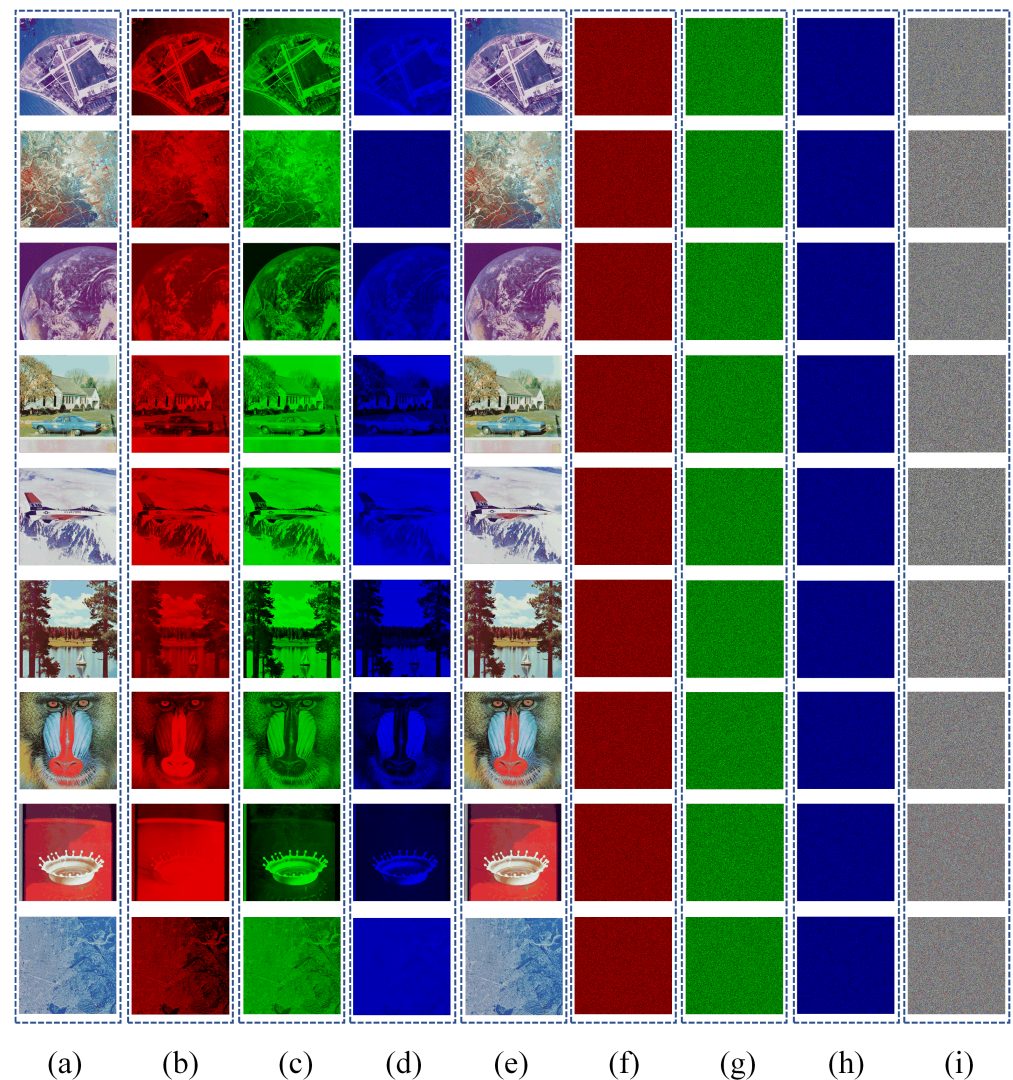


Figure 7. Encryption and decryption results: (a) nine original images; (b) nine decrypted images of red channel; (c) nine decrypted images of green channel; (d) nine decrypted images of blue channel; (e) nine decrypted images; (f) nine encrypted images of red channel; (g) nine encrypted images of green channel; (h) nine encrypted images of blue channel; (i) nine encrypted images.

Table 1. Variances of image “Jetplane” in three channels.

Channel	Variance of the Original “Jetplane”	Variance of the Encrypted “Jetplane”
Red	1,473,469	262,656
Green	1,804,092	262,656
Blue	3,220,626	262,670

The comparison between the histograms of the original “Jetplane” and the encrypted “Jetplane” is shown in Figure 8. The histograms depicted in (b), (d), and (f) indicate that the histogram of the encrypted “Jetplane” exhibits a closer resemblance to a uniform distribution. Consequently, the analysis suggests the effectiveness of the proposed encryption scheme.

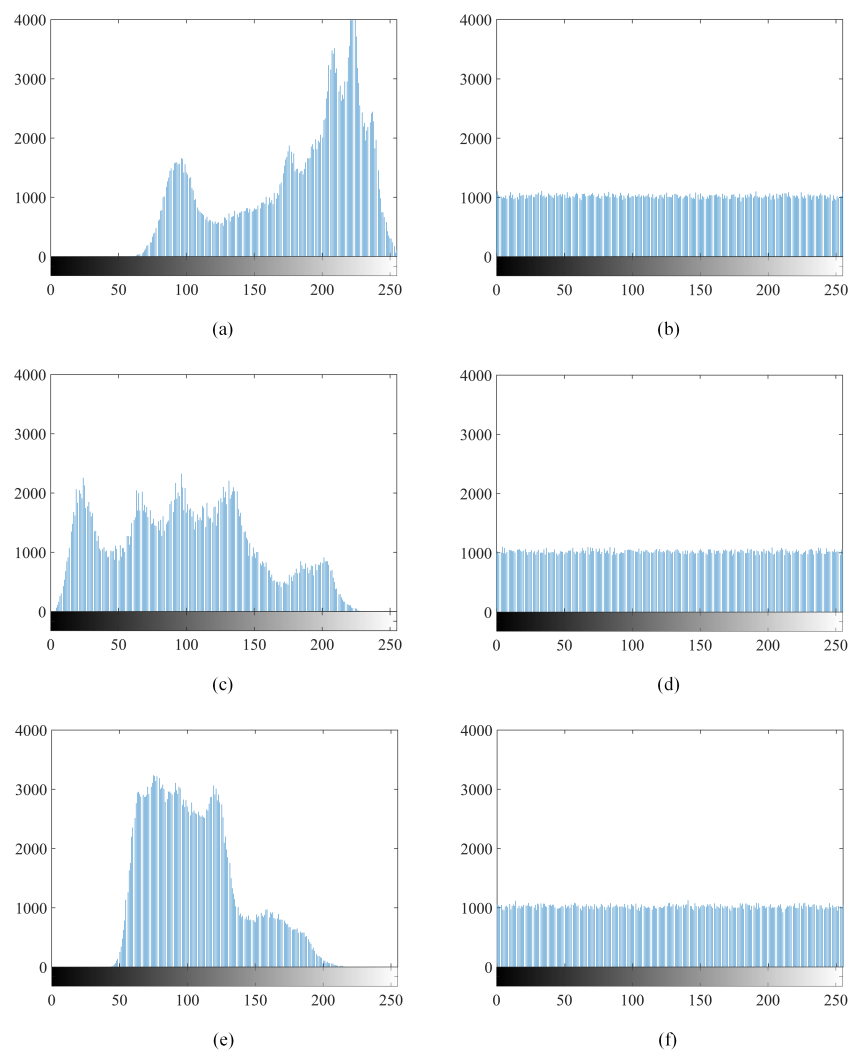


Figure 8. Comparison of the histograms: (a) the distribution of pixels within red channel of original “Jetplane”; (b) the distribution of pixels within red channel of encrypted “Jetplane”; (c) the distribution of pixels within green channel of original “Jetplane”; (d) the distribution of pixels within green channel of encrypted “Jetplane”; (e) the distribution of pixels within blue channel of original “Jetplane”; (f) the distribution of pixels within blue channel of encrypted “Jetplane”.

5.3. Key Space Analysis

In image encryption, the term “key space” denotes the collection of all keys that can be employed in the encryption process. This concept is instrumental in assessing the security

level and the quantity of keys associated with a particular encryption method. An effective encryption scheme is characterized by a substantial key space, as this feature enhances the complexity of discovering a valid key through brute force. A key space of significant magnitude is imperative for safeguarding the security integrity of the encryption scheme.

If the resolution is determined to be 10^{15} , the range of the attraction domain for variable x_1 is $x_1 \in [-0.3135, -0.2143]$. There are $0.0992 \times 10^{15} = 0.0992 \times 10^{15}$ kinds of choices. For the variables ranging from x_2 to x_{10} , the number of choices available is outlined in Table 2.

Table 2. Variables and the quantity of choices.

Variables	Quantity of Choices
x_2	0.0094×10^{15}
x_3	0.0013×10^{15}
x_4	0.0005×10^{15}
x_5	0.6329×10^{15}
x_6	2.2924×10^{15}
x_7	1.5574×10^{15}
x_8	1.9437×10^{15}
x_9	1.0475×10^{15}
x_{10}	7.0864×10^{15}

The control variables contribute to the formation of a key space, which is $0.0992 \times 10^{15} \times 0.0094 \times 10^{15} \times 0.0013 \times 10^{15} \times 0.0005 \times 10^{15} \times 0.6329 \times 10^{15} \times 2.2924 \times 10^{15} \times 1.5574 \times 10^{15} \times 1.9437 \times 10^{15} \times 1.0475 \times 10^{15} \times 7.0864 \times 10^{15} = 1.976 \times 10^{142}$. By focusing on the control variable c in Equation (3), it becomes feasible to ascertain the key space. It is $1.976 \times 10^{142} \times 2.88 \times 10^{15} = 5.691 \times 10^{157}$. When analyzing the hyperchaotic system and only considering first-order terms with a coefficient of 1, the phase space is estimated to be 6.4×10^{432} . The aforementioned computation demonstrates the phase space following a single encryption iteration. Moreover, given that this encryption scheme entails two encryption rounds, the phase space is expected to be expanded twofold. The key space expands to $6.4 \times 10^{432} \times 6.4 \times 10^{432} = 4.096 \times 10^{865} \approx 2^{877}$. It is worth noting that the calculations for key space above are not comprehensive. The effective key space of this system exceeds 2^{877} by a significant margin.

Table 3 compares the key space of the image encryption scheme proposed in this study with various other encryption methods, and the encryption scheme proposed in this paper exhibits the largest key space. The analysis shows that the proposed encryption scheme has a large enough key space to resist brute force attacks effectively.

Table 3. Comparison of key space.

Encryption Scheme	Key Space
Proposed Encryption Scheme	$>2^{877}$
Reference [35]	$>2^{116}$
Reference [46]	$>2^{478}$
Reference [47]	$>2^{219}$
Reference [48]	$>2^{604}$
Reference [49]	$>2^{554}$
Reference [50]	$>2^{372}$

5.4. Correlation Analysis

Natural images exhibit a significant level of correlation among pixels, whether they are oriented vertically, horizontally, or diagonally. A successful encryption scheme should guarantee that the encrypted image, which has decreased pixel correlation, is resistant against encryption attacks that rely on statistical analysis [51]. Adjacent pixel correlation refers to the quantification of the association among pixel values that are in close proximity to each other within an image. A low correlation can be considered to be an indication of the effectiveness of the encryption scheme. The correlation among pixels in the encrypted image tends towards zero, suggesting an enhancement in the randomness and independence of pixel values. Consequently, encrypted images display a seemingly random distribution of pixels, thereby thwarting attempts by attackers to decrypt it through basic statistical analyses or pattern recognition techniques.

The correlation is determined through the utilization of the subsequent mathematical Equations [52]:

$$\left\{ \begin{array}{l} E(m) = \frac{1}{M} \sum_{i=1}^M m_i \\ D(m) = \frac{1}{M} \sum_{i=1}^M (m_i - E(m))^2 \\ \text{cov}(m, n) = \frac{1}{M} \sum_{i=1}^M (x_i - E(m))(y_i - E(n)) \\ r_{xy} = \frac{\text{cov}(m, n)}{\sqrt{D(m) \cdot D(n)}} \end{array} \right. \quad (10)$$

Table 4 below displays the outcomes of a correlation analysis conducted on pixel values from eight images. The data in Table 4 indicate that correlation value of encrypted images approaches zero, suggesting a minimal correlation among individual pixel values within images. This lack of correlation is ascribed to the disturbance in the pixel arrangement of the original image, leading to the encrypted image manifesting as random noise. Furthermore, Table 5 provides a comparative analysis of correlation values obtained from the proposed encryption scheme in contrast to other encryption schemes. The comparison indicates that the encryption performance of the proposed scheme is similar to that of existing methods in relation to correlation. Therefore, the proposed encryption scheme demonstrates a significant degree of effectiveness.

Table 4. Correlation coefficients among eight images.

Image	Original Image			1st Round Encryption			2nd Round Encryption		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Monkey	R 0.9123	0.8625	0.8505	0.0045	0.0004	−0.0002	−0.0001	−0.0016	−0.0001
	G 0.8628	0.7811	0.7496	−0.0003	−0.0045	0.0045	−0.0037	−0.0008	−0.0014
	B 0.8965	0.8712	0.8314	−0.0001	−0.0021	0.0016	−0.0051	−0.0046	−0.0018
Mountain	R 0.8417	0.8556	0.7818	−0.0005	0.0006	−0.0022	0.0029	−0.0026	0.0022
	G 0.7543	0.7802	0.6769	0.0018	0.0007	−0.0005	−0.0010	−0.0029	0.0022
	B 0.6754	0.7083	0.5842	0.0015	0.0032	−0.0023	0.0008	0.0030	0.0035
Ice	R 0.9194	0.8997	0.8538	0.0012	0.0002	−0.0042	−0.0001	0.0001	−0.0012
	G 0.9189	0.9102	0.8596	−0.0033	−0.0018	−0.0013	0.0006	0.0028	−0.0016
	B 0.8957	0.8888	0.8367	−0.0017	−0.0001	0.0017	0.0015	0.0049	−0.0024
Apartment	R 0.9552	0.9591	0.9252	−0.0008	0.0005	−0.0003	0.0002	−0.0032	0.0015
	G 0.9405	0.9445	0.8951	−0.0001	0.0011	0.0014	0.0005	0.0030	0.0021
	B 0.9728	0.9691	0.9456	0.0032	−0.0008	−0.0032	−0.0004	0.0022	0.0019

Table 4. Cont.

Image	Original Image			1st Round Encryption			2nd Round Encryption		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lake	R 0.9713	0.9697	0.9534	0.0018	−0.0004	0.0009	0.0021	0.0014	−0.0014
	G 0.9718	0.9666	0.9534	−0.0005	0.0003	−0.0002	0.0029	0.0020	0.0011
	B 0.9713	0.9697	0.9534	−0.0022	0.0010	0.0010	0.0015	−0.0006	0.0037
spatter	R 0.9938	0.9953	0.9898	0.0019	0.0008	−0.0006	0.0017	−0.0005	−0.0040
	G 0.9812	0.9872	0.9713	0.0019	0.0008	−0.0006	0.0031	−0.0047	0.0022
	B 0.9826	0.9792	0.9653	0.0004	0.0007	−0.0024	0.0040	−0.0024	−0.0004
Earth	R 0.9695	0.9737	0.9518	−0.0013	−0.0048	0.0020	−0.0001	−0.0012	0.0024
	G 0.9683	0.9734	0.9492	0.0022	−0.0012	0.0013	−0.0002	−0.0033	−0.0004
	B 0.9572	0.9623	0.9337	0.0016	0.0012	−0.0002	0.0041	−0.0019	−0.0049
Jetplane	R 0.9738	0.9593	0.9382	0.0019	0.0019	0.0019	−0.0035	0.0023	−0.0022
	G 0.9596	0.9691	0.9356	0.0005	0.0006	0.0013	−0.0036	−0.0010	0.0016
	B 0.9673	0.9431	0.9249	−0.0006	0.0001	0.0019	0.0007	0.0011	0.0004

Table 5. Comparison in terms of pixel correlation.

Encryption Scheme	Horizontal	Vertical	Diagonal
Proposed Encryption Scheme	0.0013	−0.0021	−0.0010
Reference [35]	−0.0093	0.0025	−0.0024
Reference [53]	−0.0012	0.0099	−0.0032
Reference [54]	0.0027	0.0015	0.0019
Reference [55]	0.0023	−0.0010	0.0009

As indicated in Table 4, the original image exhibits a strong correlation coefficient, whereas the encrypted image demonstrates a correlation coefficient that approaches zero [56].

The analysis of pixel correlations in the image is performed in vertical, horizontal, and diagonal directions using the image “Apartment” as an example. The original “Apartment” image, the first round encryption of the “Apartment” image, and the second round encryption of the “Apartment” image are depicted in Figure 9, Figure 10 and Figure 11, respectively. Figures 10 and 11 illustrate the outcomes of selecting every sixth point at regular intervals. The original “Apartment” image, as shown in Figure 9, exhibits a high degree of correlation among pixels in the vertical, horizontal, and diagonal orientations, with a predominant alignment along the $y = x$ line. Conversely, Figures 10 and 11 reveal that pixel points of the encrypted “Apartment” image are dispersed throughout the entire image area, indicating a diminished correlation between pixels. The comparison highlights a significantly reduced correlation between pixels in the encrypted “Apartment” image. This prevents attackers from reconstructing the original image by analyzing the relationships between pixels.

5.5. Information Entropy Analysis

Information entropy is frequently used in image encryption to measure the amount of uncertainty related to information. Within image encryption, entropy is a measure of the disorder of image pixels. Enhanced encryption performance is typically associated with a higher level of disorder among pixels in the encrypted image. Consequently, in image encryption, information entropy of an encrypted image is expected to closely approach the maximum value, which is 8 [57]. Information entropy can be determined using the subsequent equation:

$$H(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (11)$$

In Equation (11), $H(X)$ represents the information entropy of the random variable X . The probability $p(x_i)$ represents the probability of the random variable X assuming the value x_i . Table 6 compares the information entropy of original and encrypted images for eight images, showing that the information entropy of the encrypted images closely approximates the expected theoretical value. Table 7 contrasts the information entropy of encrypted images utilizing various encryption methods. The information entropy associated with the proposed encryption scheme aligns closely with the optimal value of 8, surpassing the outcomes of other encryption methods. Consequently, the proposed encryption scheme is deemed to possess a high level of security.

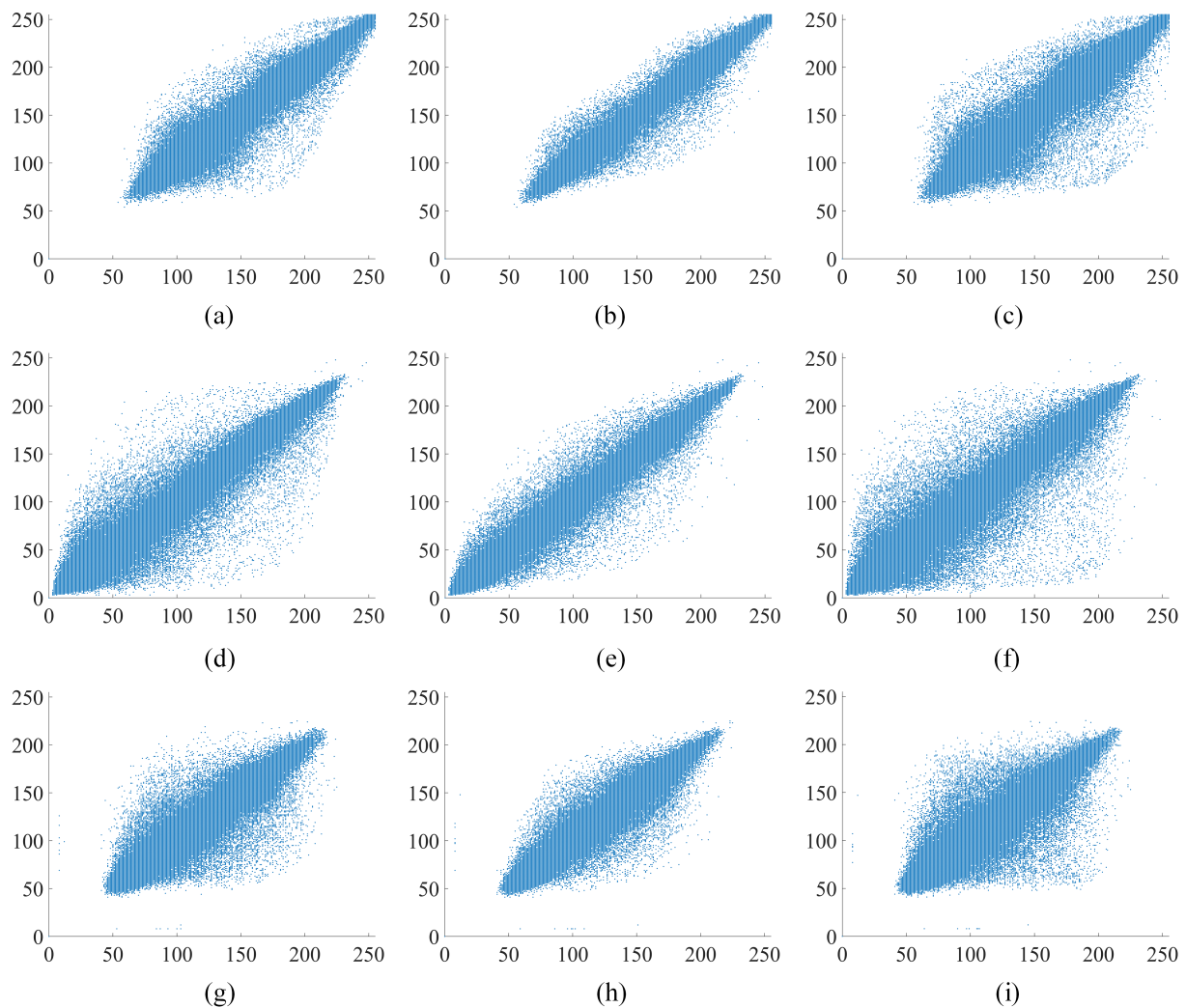


Figure 9. Analysis of correlation in the original “Apartment” image: (a) the horizontal orientation within red channel of “Apartment”; (b) the vertical orientation within red channel of “Apartment”; (c) the diagonal orientation within red channel of “Apartment”; (d) the horizontal orientation within green channel of “Apartment”; (e) the vertical orientation within green channel of “Apartment”; (f) the diagonal orientation within green channel of “Apartment”; (g) the horizontal orientation within blue channel of “Apartment”; (h) the vertical orientation within blue channel of “Apartment”; (i) the diagonal orientation within blue channel of “Apartment”.

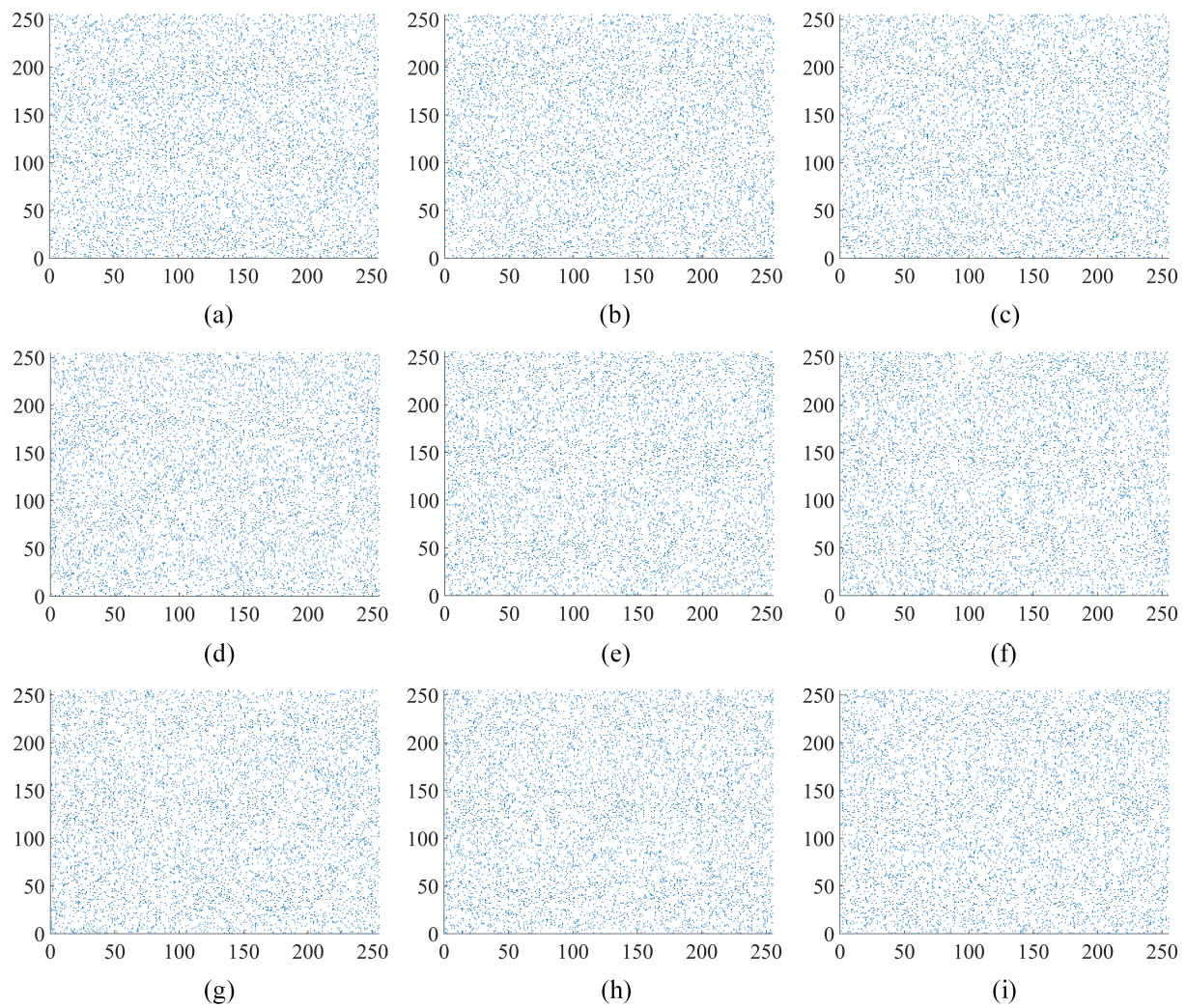


Figure 10. Analysis of correlation after the first round of encryption in “Apartment”: (a) the horizontal orientation within red channel of “Apartment”; (b) the vertical orientation within red channel of “Apartment”; (c) the diagonal orientation within red channel of “Apartment”; (d) the horizontal orientation within green channel of “Apartment”; (e) the vertical orientation within green channel of “Apartment”; (f) the diagonal orientation within green channel of “Apartment”; (g) the horizontal orientation within blue channel of “Apartment”; (h) the vertical orientation within blue channel of “Apartment”; (i) the diagonal orientation within blue channel of “Apartment”.

Table 6. Information entropy of eight images.

Image	Original Image	1st Round Encrypted	2nd Round Encrypted	Ideal Value
Monkey	7.1073	7.9998	7.9998	8
Radar	7.5787	7.9998	7.9998	8
Ice	7.3983	7.9997	7.9997	8
Apartment	7.4858	7.9998	7.9998	8
Lake	7.7622	7.9998	7.9998	8
spatter	7.2428	7.9998	7.9998	8
Earth	7.3983	7.9997	7.9997	8
Jetplane	6.6639	7.9997	7.9997	8

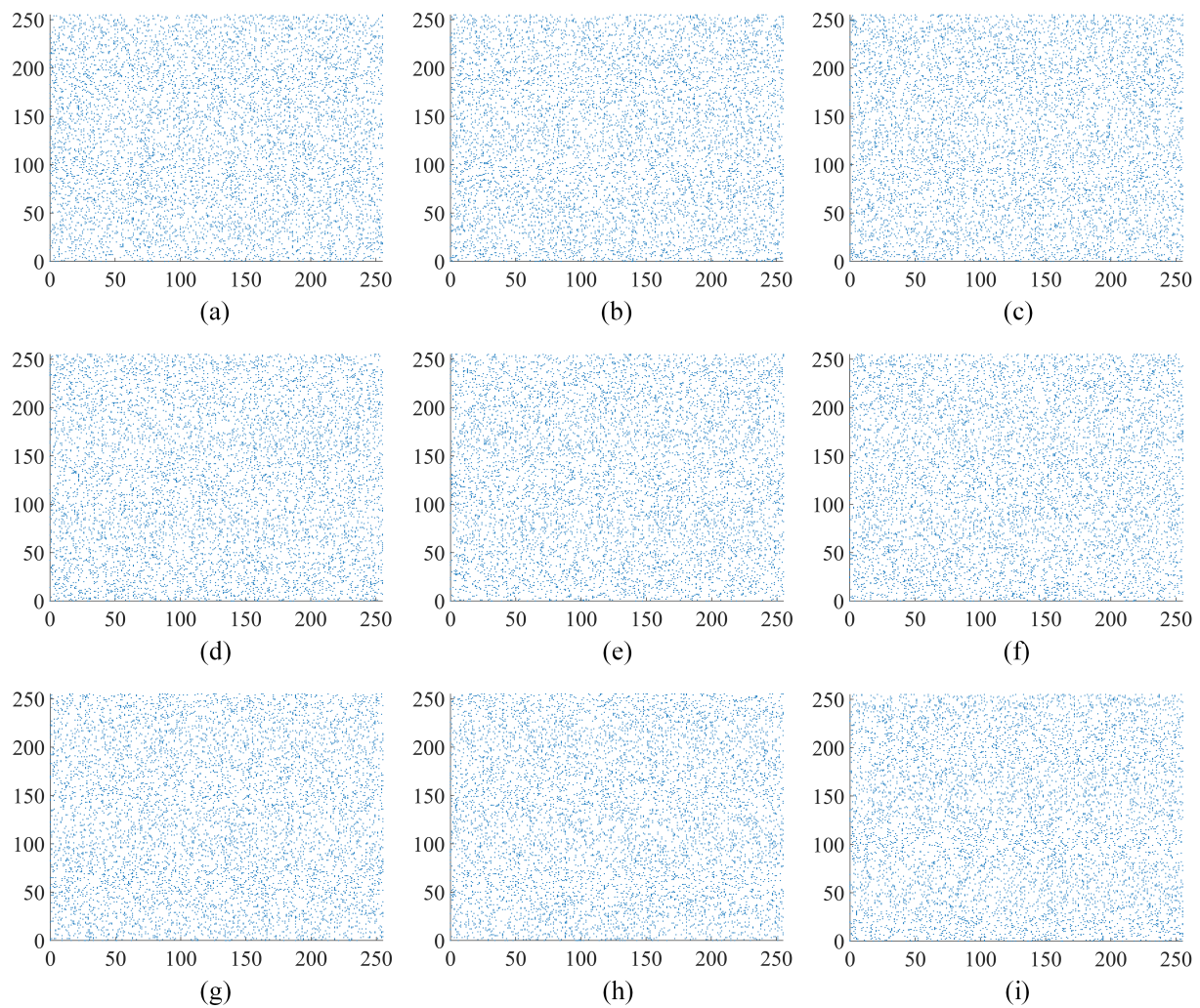


Figure 11. Analysis of correlation after the second round encryption in “Apartment”: (a) the horizontal orientation within red channel of “Apartment”; (b) the vertical orientation within red channel of “Apartment”; (c) the diagonal orientation within red channel of “Apartment”; (d) the horizontal orientation within green channel of “Apartment”; (e) the vertical orientation within green channel of “Apartment”; (f) the diagonal orientation within green channel of “Apartment”; (g) the horizontal orientation within blue channel of “Apartment”; (h) the vertical orientation within blue channel of “Apartment”; (i) the diagonal orientation within blue channel of “Apartment”.

Table 7. Comparison in terms of information entropy.

Encryption Scheme	Information Entropy
Proposed Encryption Scheme	7.9998
Reference [35]	7.9993
Reference [58]	7.9993
Reference [50]	7.9821
Reference [48]	7.9971
Reference [59]	7.9966

5.6. Key Sensitivity Analysis

The term “key sensitivity” refers to the phenomenon where slight alterations in the initial key can result in notable variations in the keys produced by the iterative function.

A minor adjustment in the key can lead to a substantial transformation in encrypted images. Key sensitivity is a crucial feature in image encryption to guarantee data security, and it is used to assess the importance of an encryption scheme.

In order to illustrate the importance of sensitivity, we conducted an experiment where we encrypted an image using keys that were only slightly different. Specifically, we introduced a minute increment of 10^{-14} to the original key during this process. Figure 12 illustrates the waveform for c in Equation (3) and $c + \Delta c$, along with their changing difference over time. After subtracting images encrypted with different keys, it is apparent from the data presented in Figure 13 that the resulting images display significant differences.

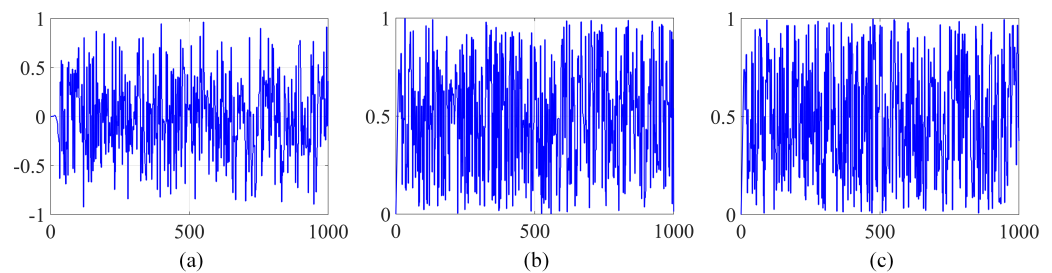


Figure 12. (a) the value of Δc ; (b) the value of $c + \Delta c$; (c) the value of c .

Key sensitivity analysis in image encryption schemes is commonly assessed using two main metrics: NPCR and UACI. The two metrics are crucial in evaluating the effectiveness and robustness of image encryption methods. These metrics are formally defined as part of the evaluation process. These indicators are defined as [60]:

$$\text{NPCR} = \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M \delta_{i,j} \quad (12)$$

$$\text{UACI} = \frac{1}{255 \times N \times M} \sum_{i=1}^N \sum_{j=1}^M |E_{i,j} - F_{i,j}| \quad (13)$$

The variables M and N correspond to the width and height of the image, and $\delta_{i,j}$ signifies the alteration in pixel intensity at coordinates (i, j) . $E_{i,j}$ is the pixel value of the original image at the coordinates (i, j) , while $F_{i,j}$ is the pixel value of the encrypted image at the same location.

The theoretical values for UACI and NPCR are reported as 33.4635% and 99.6043%, respectively [61]. The results presented in Tables 8 and 9 demonstrate a high level of concordance with the theoretical values, suggesting robust key sensitivity in the proposed encryption scheme.

Table 8. Analysis of the key sensitivity in terms of NPCR.

Image	NPCR (%) Ideal Value: 99.60		
	R	G	B
Monkey	99.5972	99.5983	99.6029
Mountain	99.6017	99.6075	99.6159
Boat	99.6006	99.5960	99.5972
Apartment	99.6132	99.6143	99.6044
Lake	99.5953	99.6071	99.6140
spatter	99.5934	99.5934	99.6086
Earth	99.5934	99.6123	99.5934
Jetplane	99.6140	99.6158	99.6075

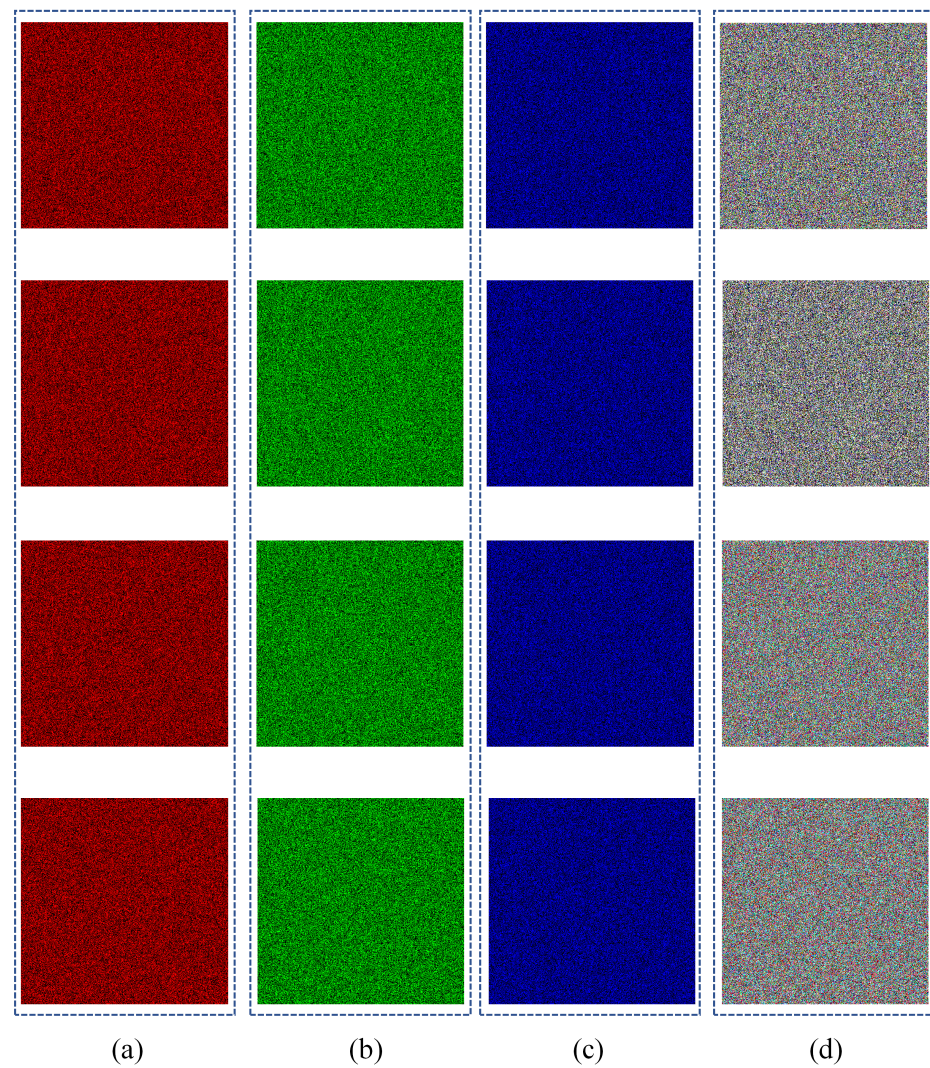


Figure 13. Analysis of key sensitivity: (a) the discrepancy within red channel; (b) the discrepancy within green channel; (c) the discrepancy within blue channel; (d) the discrepancy within three channels.

Table 9. Analysis of the key sensitivity in terms of UACI.

Image	UACI (%) Ideal Value: 33.46 (%)		
	R	G	B
Monkey	33.4656	33.4756	33.4610
Mountain	33.4671	33.4670	33.4788
Boat	33.4581	33.4607	33.4779
Apartment	33.4675	33.4680	33.4658
Lake	33.4721	33.4619	33.4553
spatter	33.4695	33.4635	33.4720
Earth	33.4643	33.4529	33.4711
Jetplane	33.4677	33.4611	33.4769

5.7. Differential Attack

Differential attack refers to an attack method that utilizes the differences between pixels in an image to crack the encryption scheme. Aiming to mitigate risks of differential attacks, robust encryption methods should demonstrate a heightened response to variations in plaintext. Even a slight alteration in the pixel values of the original image can cause significant changes in a generated encrypted image. In this study, encryption was performed by adjusting the top-left corner pixel value in the original image “Boat”. The analysis revealed a notable difference between the resulting encrypted “Boat” image and the original encrypted “Boat” image. The outcomes are illustrated in Figure 14.

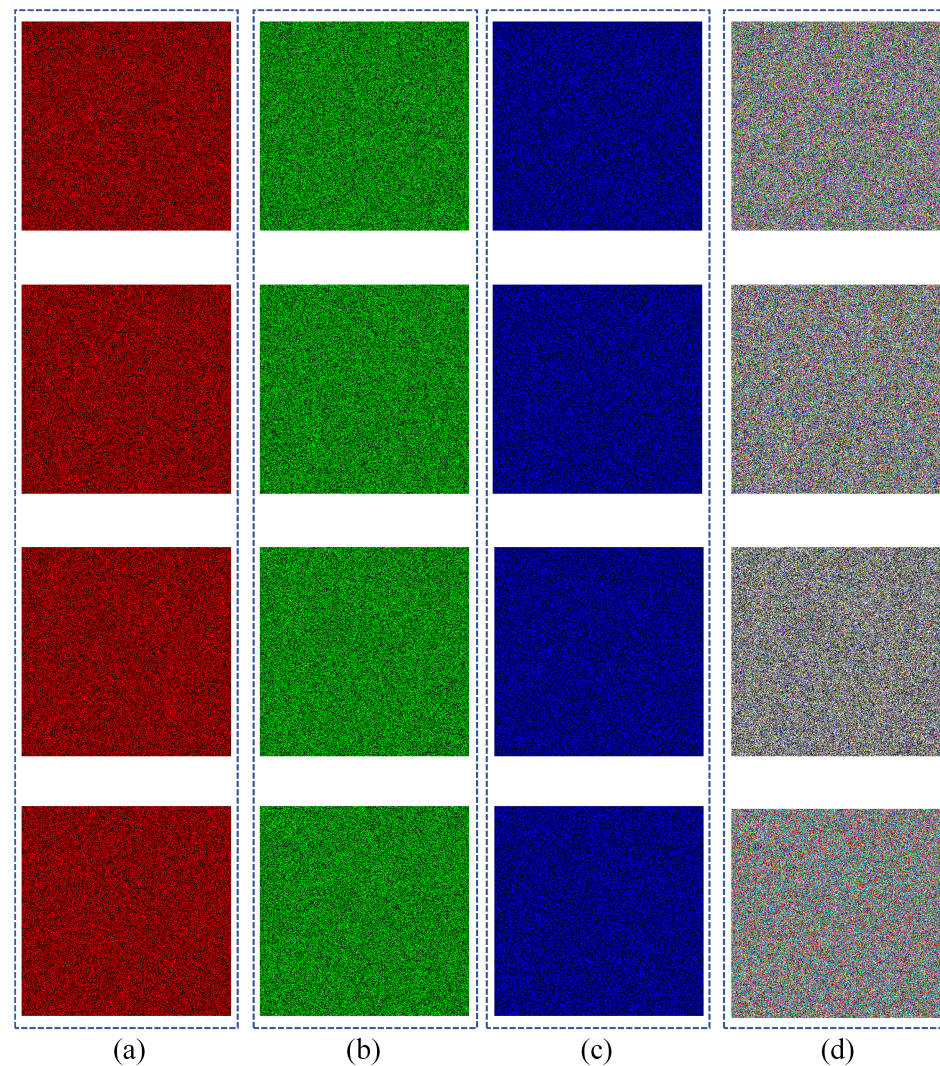


Figure 14. Analysis of differential attack: (a) the difference within red channel; (b) the difference within green channel; (c) the difference within blue channel; (d) the difference within three channels.

The NPCR and UACI of differential attacks are detailed in Table 10 and Table 11, respectively. Table 12 compares the UACI and NPCR values calculated from the proposed encryption scheme with those obtained from other encryption schemes. The NPCR and UACI values presented in Tables 10 and 11 closely approximate the ideal values, indicating that the proposed encryption scheme is proficient in mitigating against differential attacks.

Table 10. Analysis of the differential attack in terms of NPCR.

Image	NPCR (%) Ideal Value: 99.60			
	R	G	B	Average
Monkey	99.5998	99.5961	99.6127	99.6029
Radar	99.5930	99.6177	99.6159	99.6089
Ice	99.6181	99.6071	99.6147	99.6133
Apartment	99.6155	99.6044	99.6108	99.6102
Lake	99.6113	99.6189	99.6075	99.6126
spatter	99.6181	99.6132	99.6059	99.6124
Earth	99.6197	99.6168	99.5964	99.6110
Jetplane	99.6154	99.6107	99.5995	99.6085

Table 11. Analysis of the differential attack in terms of UACI.

Image	UACI (%) Ideal Value: 33.46			
	R	G	B	Average
Monkey	33.4645	33.4675	33.4741	33.4687
Radar	33.4694	33.4647	33.4638	33.4660
Ice	33.4727	33.4637	33.4728	33.4697
Apartment	33.4692	33.4623	33.4751	33.4689
Lake	33.4638	33.4687	33.4747	33.4691
spatter	33.4637	33.4698	33.4742	33.4692
Earth	33.4786	33.4765	33.4781	33.4777
Jetplane	33.4665	33.4673	33.4673	33.4670

Table 12. Comparison of NPCR and UACI in terms of differential attack.

Encryption Scheme	NPCR (%)	UACI (%)
Proposed Encryption Scheme	99.6100	33.4695
Reference [35]	99.6065	33.4595
Reference [55]	99.5320	33.4500
Reference [62]	99.5855	30.3873

5.8. Cropping Attack

During the data transfer process, the encrypted image may become corrupted. A robust encryption scheme should be capable of withstanding partial data loss during transmission. If the encrypted image is partially damaged, an excellent encryption should recover the image as much as possible through decryption [63].

Figure 15 illustrates that even with data loss rates of 12.5%, 12.5%, 25%, 37.5%, 50%, and 62.5%, the images that have been decrypted exhibit a strong resemblance to the original image, effectively preserving important information. The results indicate that the encryption scheme is capable of withstanding cropping attacks.

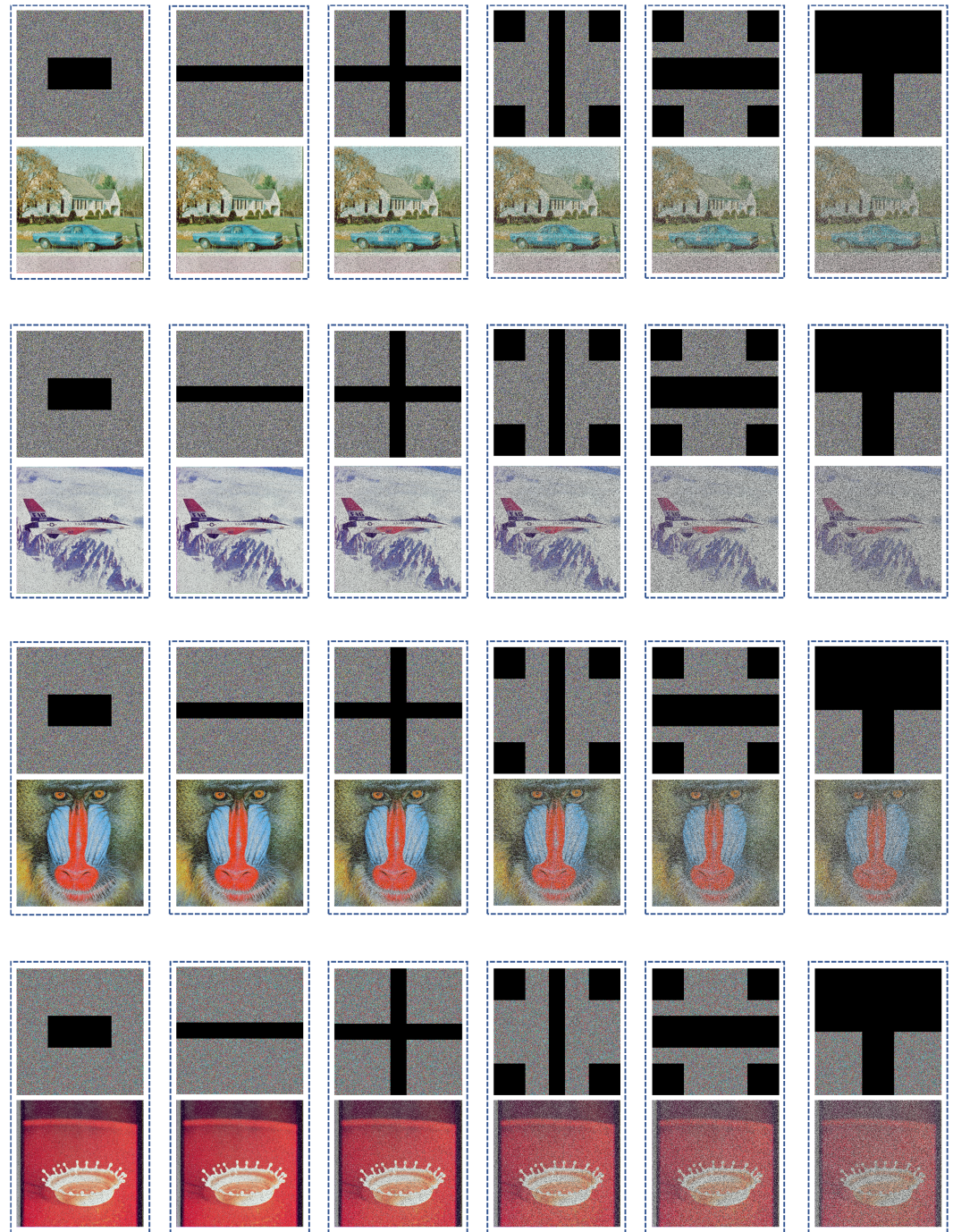


Figure 15. Analysis of cropping attacks in four different images.

5.9. Comparison with Existing Methods

In this section, a thorough evaluation is carried out to compare the proposed encryption scheme with current encryption schemes, focusing on aspects related to security and application scenarios.

In the realm of security, the encryption scheme in this paper is evaluated in comparison to established encryption schemes by examining factors such as information entropy, correlation coefficient, key space, UACI, and NPCR. The encryption scheme demonstrates superior performance, as evidenced by the preceding evaluation and comparison.

The encryption scheme is applicable to the medical field, as evidenced by Liu et al. in their utilization of a chaos-based encryption method for medical images [64]. Encryption schemes intended for medical applications are required to adhere to specific criteria, such as

ensuring robust security measures and maintaining data integrity. The proposed encryption scheme exhibits potential for application in medical contexts, as indicated by the assessment of factors including histograms, key space, key sensitivity, and pixel correlation.

6. Conclusions

In this study, a novel color image encryption scheme that employs a 4D hyperchaotic system is presented. The scheme involves the generation of four chaotic sequences for two encryption rounds, thereby enhancing the pseudorandomness and unpredictability of the encrypted images. The hyperchaotic system utilized in this method outperforms low-dimensional chaotic systems in terms of key space efficiency. Moreover, in comparison to existing encryption methods, the proposed scheme offers a larger key space, thereby improving its resistance against both differential and cropping attacks. Evaluation metrics, including entropy, pixel correlation, NPCR, and UACI values, demonstrate a close alignment of the encrypted images with ideal values. The encryption scheme demonstrates notable effectiveness in preserving the security of color images.

Author Contributions: Conceptualization, J.L. and Z.L.; methodology, J.L.; software, H.Q.; validation, Z.L., J.L. and H.Q.; formal analysis, X.Z.; investigation, J.Y.; resources, J.L.; data curation, Z.L.; writing—original draft preparation, Z.L.; writing—review and editing, J.Y.; visualization, J.L.; supervision, H.Q.; project administration, J.Y.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Regional Project of the National Natural Science Foundation of China grant number 82260364. The APC was funded by the Gansu Provincial Science and Technology Department grant numbers 22JR5RA166 and 22JR5RA555, and the Gansu Higher Education Innovation Fund Project grant number 2022B-084.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article. The data underpinning the conclusions of this research are accessible from the corresponding author upon a justifiable request.

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

References

1. Ji, X.; Dong, Z.; Han, Y.; Lai, C.S.; Zhou, G.; Qi, D. EMSN: An Energy-Efficient Memristive Sequencer Network for Human Emotion Classification in Mental Health Monitoring. *IEEE Trans. Consum. Electron.* **2023**, *69*, 1005–1016. [\[CrossRef\]](#)
2. Yang, Z.; Lian, J.; Liu, J. Infrared UAV Target Detection Based on Continuous-Coupled Neural Network. *Micromachines* **2023**, *14*, 2113. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Zheng, Y.; Yu, Z.; Wang, S.; Huang, T. Spike-based motion estimation for object tracking through bio-inspired unsupervised learning. *IEEE Trans. Image Process.* **2022**, *32*, 335–349. [\[CrossRef\]](#)
4. Li, Y.; Fang, X.; Gao, Y.; Zhou, D.; Shen, J.; Liu, J.K.; Pan, G.; Xu, Q. Efficient Structure Slimming for Spiking Neural Networks. *IEEE Trans. Artif. Intell.* **2024**, 1–9. [\[CrossRef\]](#)
5. Zhu, H.; Liu, C.; Wu, W.Z.; Xie, W.; Lao, T. Weakened fractional-order accumulation operator for ill-conditioned discrete grey system models. *Appl. Math. Model.* **2022**, *111*, 349–362. [\[CrossRef\]](#)
6. Yao, Z.; Sun, K.; He, S. Firing patterns in a fractional-order FithzHugh–Nagumo neuron model. *Nonlinear Dyn.* **2022**, *110*, 1807–1822. [\[CrossRef\]](#)
7. Mou, J.; Shin, D.H.; Cohen, J. Understanding trust and perceived usefulness in the consumer acceptance of an e-service: A longitudinal investigation. *Behav. Inf. Technol.* **2017**, *36*, 125–139. [\[CrossRef\]](#)
8. Liu, J.; Lian, J.; Sprott, J.C.; Liu, Q.; Ma, Y. The butterfly effect in primary visual cortex. *IEEE Trans. Comput.* **2022**, *71*, 2803–2815. [\[CrossRef\]](#)
9. Min, F.; Zhai, G.; Yin, S.; Zhong, J. Switching bifurcation of a Rulkov neuron system with ReLu-type memristor. *Nonlinear Dyn.* **2024**, *112*, 5687–5706. [\[CrossRef\]](#)
10. Cheng, Y.; Min, F. Symmetry, Multistability and Antimonotonicity of a Shinriki Oscillator with Dual Memristors. *Int. J. Bifurc. Chaos* **2023**, *33*, 2350186. [\[CrossRef\]](#)
11. Xu, Q.; Shen, J.; Ran, X.; Tang, H.; Pan, G.; Liu, J.K. Robust transcoding sensory information with neural spikes. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 1935–1946. [\[CrossRef\]](#)

12. Zhang, M.; Wang, J.; Wu, J.; Belatreche, A.; Amornpaisannon, B.; Zhang, Z.; Miriyala, V.P.K.; Qu, H.; Chua, Y.; Carlson, T.E.; et al. Rectified Linear Postsynaptic Potential Function for Backpropagation in Deep Spiking Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1947–1958. [[CrossRef](#)] [[PubMed](#)]
13. Wu, J.; Xu, C.; Han, X.; Zhou, D.; Zhang, M.; Li, H.; Tan, K.C. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7824–7840. [[CrossRef](#)] [[PubMed](#)]
14. Guo, M.; Yang, R.; Zhang, M.; Liu, R.; Zhu, Y.; Dou, G. A novel memcapacitor and its application in a chaotic circuit. *Nonlinear Dyn.* **2021**, *105*, 877–886. [[CrossRef](#)]
15. Guo, M.; Xue, Y.; Gao, Z.; Zhang, Y.; Dou, G.; Li, Y. Dynamic analysis of a physical SBT memristor-based chaotic circuit. *Int. J. Bifurc. Chaos* **2017**, *27*, 1730047. [[CrossRef](#)]
16. Li, H.L.; Cao, J.; Hu, C.; Jiang, H.; Alsaadi, F.E. Synchronization Analysis of Discrete-Time Fractional-Order Quaternion-Valued Uncertain Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–12. [[CrossRef](#)] [[PubMed](#)]
17. Zhu, H.; Dai, L.; Liu, Y.; Wu, L. A three-dimensional bit-level image encryption algorithm with Rubik's cube method. *Math. Comput. Simul.* **2021**, *185*, 754–770. [[CrossRef](#)]
18. Wang, J.; Lian, J.; Ma, M.; Lei, J.; Li, C.; Li, B.; Liu, J. Chaos in Motion: Unveiling Robustness in Remote Heart Rate Measurement through Brain-Inspired Skin Tracking. *arXiv* **2024**, arXiv:2404.07687.
19. Niyat, A.Y.; Moattar, M.H.; Torshiz, M.N. Color image encryption based on hybrid hyper-chaotic system and cellular automata. *Opt. Lasers Eng.* **2017**, *90*, 225–237. [[CrossRef](#)]
20. Kumar, A.; Dua, M. Novel pseudo random key & cosine transformed chaotic maps based satellite image encryption. *Multimed. Tools Appl.* **2021**, *80*, 27785–27805.
21. Norouzi, B.; Mirzakuchaki, S. A fast color image encryption algorithm based on hyper-chaotic systems. *Nonlinear Dyn.* **2014**, *78*, 995–1015. [[CrossRef](#)]
22. Kumar, A.; Dua, M. A GRU and chaos-based novel image encryption approach for transport images. *Multimed. Tools Appl.* **2023**, *82*, 18381–18408. [[CrossRef](#)]
23. Zhang, X.; Sun, L.; Geng, X.; Yue, H.; Zhao, X.; Lei, J.; Liu, J. A novel image encryption scheme based on ccnn. *Phys. Scr.* **2024**, *99*, 025253. [[CrossRef](#)]
24. Liu, J.; Tang, S.; Lian, J.; Ma, Y.; Zhang, X. A novel fourth order chaotic system and its algorithm for medical image encryption. *Multidimens. Syst. Signal Process.* **2019**, *30*, 1637–1657. [[CrossRef](#)]
25. Zhu, H.; Ge, J.; Qi, W.; Zhang, X.; Lu, X. Dynamic analysis and image encryption application of a sinusoidal-polynomial composite chaotic system. *Math. Comput. Simul.* **2022**, *198*, 188–210. [[CrossRef](#)]
26. Gao, X.; Mou, J.; Xiong, L.; Sha, Y.; Yan, H.; Cao, Y. A fast and efficient multiple images encryption based on single-channel encryption and chaotic system. *Nonlinear Dyn.* **2022**, *108*, 613–636. [[CrossRef](#)]
27. Zhu, W.; Sun, K.; He, S.; Wang, H.; Liu, W. A class of m-dimension grid multi-cavity hyperchaotic maps and its application. *Chaos Solitons Fractals* **2023**, *170*, 113370. [[CrossRef](#)]
28. Jin, M.; Sun, K.; He, S. A novel fractional-order hyperchaotic complex system and its synchronization. *Chin. Phys. B* **2023**, *32*, 060501. [[CrossRef](#)]
29. Wang, X.; Gao, M.; Iu, H.H.C.; Wang, C. Tri-valued memristor-based hyper-chaotic system with hidden and coexistent attractors. *Chaos Solitons Fractals* **2022**, *159*, 112177. [[CrossRef](#)]
30. Li, Y.; Li, C.; Lei, T.; Yang, Y.; Chen, G. Offset Boosting-Entangled Complex Dynamics in the Memristive Rulkov Neuron. *IEEE Trans. Ind. Electron.* **2024**, *71*, 9569–9579. [[CrossRef](#)]
31. Shang, C.; He, S.; Rajagopal, K.; Wang, H.; Sun, K. Dynamics and chimera state in a neural network with discrete memristor coupling. *Eur. Phys. J. Spec. Top.* **2022**, *231*, 4065–4076. [[CrossRef](#)]
32. 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](#)]
33. Yang, F.; Mou, J.; Sun, K.; Cao, Y.; Jin, J. Color image compression-encryption algorithm based on fractional-order memristor chaotic circuit. *IEEE Access* **2019**, *7*, 58751–58763. [[CrossRef](#)]
34. Lin, Z.; Liu, J.; Lian, J.; Ma, Y.; Zhang, X. A novel fast image encryption algorithm for embedded systems. *Multimed. Tools Appl.* **2019**, *78*, 20511–20531. [[CrossRef](#)]
35. Kamal, S.T.; Hosny, K.M.; Elgindy, T.M.; Darwish, M.M.; Fouda, M.M. A New Image Encryption Algorithm for Grey and Color Medical Images. *IEEE Access* **2021**, *9*, 37855–37865. [[CrossRef](#)]
36. Wang, X.; Zhang, X.; Gao, M.; Tian, Y.; Wang, C.; Iu, H.H.C. A color image encryption algorithm based on hash table, hilbert curve and hyper-chaotic synchronization. *Mathematics* **2023**, *11*, 567. [[CrossRef](#)]
37. Sun, K.; Liu, X.; Zhu, C.; Sprott, J. Hyperchaos and hyperchaos control of the sinusoidally forced simplified Lorenz system. *Nonlinear Dyn.* **2012**, *69*, 1383–1391. [[CrossRef](#)]
38. Bao, B.; Zhao, Q.; Yu, X.; Wu, H.; Xu, Q. Complex dynamics and initial state effects in a two-dimensional sine-bounded memristive map. *Chaos Solitons Fractals* **2023**, *173*, 113748. [[CrossRef](#)]
39. Wang, C.; Tang, D.; Lin, H.; Yu, F.; Sun, Y. High-dimensional memristive neural network and its application in commercial data encryption communication. *Expert Syst. Appl.* **2024**, *242*, 122513. [[CrossRef](#)]
40. Mahmoud, G.M.; Mahmoud, E.E.; Ahmed, M.E. On the hyperchaotic complex Lü system. *Nonlinear Dyn.* **2009**, *58*, 725–738. [[CrossRef](#)]

41. Liu, J.; Ma, J.; Lian, J.; Chang, P.; Ma, Y. An approach for the generation of an nth-order chaotic system with hyperbolic sine. *Entropy* **2018**, *20*, 230. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Yang, J.; Li, H.L.; Zhang, L.; Hu, C.; Jiang, H. Synchronization of discrete-time fractional fuzzy neural networks with delays via quantized control. *ISA Trans.* **2023**, *141*, 241–250. [\[CrossRef\]](#) [\[PubMed\]](#)
43. Singh, J.P.; Roy, B. The nature of Lyapunov exponents is (+,+, -,-). Is it a hyperchaotic system? *Chaos Solitons Fractals* **2016**, *92*, 73–85. [\[CrossRef\]](#)
44. Qiu, H.; Zhang, X.; Yue, H.; Liu, J. A Novel Eighth-Order Hyperchaotic System and Its Application in Image Encryption. *Mathematics* **2023**, *11*, 4099. [\[CrossRef\]](#)
45. Li, X.; Zeng, J.; Ding, Q.; Fan, C. A novel color image encryption algorithm based on 5-D hyperchaotic system and DNA sequence. *Entropy* **2022**, *24*, 1270. [\[CrossRef\]](#) [\[PubMed\]](#)
46. Alexan, W.; Elkandoz, M.; Mashaly, M.; Azab, E.; Aboshousha, A. Color image encryption through chaos and kaa map. *IEEE Access* **2023**, *11*, 11541–11554. [\[CrossRef\]](#)
47. Hu, X.; Wei, L.; Chen, W.; Chen, Q.; Guo, Y. Color image encryption algorithm based on dynamic chaos and matrix convolution. *IEEE Access* **2020**, *8*, 12452–12466. [\[CrossRef\]](#)
48. Iqbal, N.; Naqvi, R.A.; Atif, M.; Khan, M.A.; Hanif, M.; Abbas, S.; Hussain, D. On the image encryption algorithm based on the chaotic system, dna encoding, and castle. *IEEE Access* **2021**, *9*, 118253–118270. [\[CrossRef\]](#)
49. Elkandoz, M.T.; Alexan, W. Image encryption based on a combination of multiple chaotic maps. *Multimed. Tools Appl.* **2022**, *81*, 25497–25518. [\[CrossRef\]](#)
50. Gabr, M.; Younis, H.; Ibrahim, M.; Alajmy, S.; Khalid, I.; Azab, E.; Elias, R.; Alexan, W. Application of dna coding, the lorenz differential equations and a variation of the logistic map in a multi-stage cryptosystem. *Symmetry* **2022**, *14*, 2559. [\[CrossRef\]](#)
51. Ye, G.; Jiao, K.; Huang, X. Quantum logistic image encryption algorithm based on SHA-3 and RSA. *Nonlinear Dyn.* **2021**, *104*, 2807–2827. [\[CrossRef\]](#)
52. Liu, S.; Ye, G. Asymmetric image encryption algorithm using a new chaotic map and an improved radial diffusion. *Optik* **2023**, *288*, 171181. [\[CrossRef\]](#)
53. Chai, X.; Gan, Z.; Yuan, K.; Chen, Y.; Liu, X. A novel image encryption scheme based on DNA sequence operations and chaotic systems. *Neural Comput. Appl.* **2019**, *31*, 219–237. [\[CrossRef\]](#)
54. Chandrasekaran, J.; Thiruvengadam, S. A hybrid chaotic and number theoretic approach for securing DICOM images. *Secur. Commun. Netw.* **2017**, *2017*, 6729896. [\[CrossRef\]](#)
55. Wang, X.; Liu, C. A novel and effective image encryption algorithm based on chaos and DNA encoding. *Multimed. Tools Appl.* **2017**, *76*, 6229–6245. [\[CrossRef\]](#)
56. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137. [\[CrossRef\]](#)
57. Zhen, P.; Zhao, G.; Min, L.; Jin, X. Chaos-based image encryption scheme combining DNA coding and entropy. *Multimed. Tools Appl.* **2016**, *75*, 6303–6319. [\[CrossRef\]](#)
58. Hua, Z.; Yi, S.; Zhou, Y. Medical image encryption using high-speed scrambling and pixel adaptive diffusion. *Signal Process.* **2018**, *144*, 134–144. [\[CrossRef\]](#)
59. Alexan, W.; ElBeltagy, M.; Aboshousha, A. Rgb image encryption through cellular automata, s-box and the lorenz system. *Symmetry* **2022**, *14*, 443. [\[CrossRef\]](#)
60. Zhou, Y.; Hua, Z.; Pun, C.M.; Philip Chen, C.L. Cascade Chaotic System with Applications. *IEEE Trans. Cybern.* **2015**, *45*, 2001–2012. [\[CrossRef\]](#) [\[PubMed\]](#)
61. Wang, M.; Wang, X.; Zhang, Y.; Zhou, S.; Zhao, T.; Yao, N. A novel chaotic system and its application in a color image cryptosystem. *Opt. Lasers Eng.* **2019**, *121*, 479–494. [\[CrossRef\]](#)
62. Alexan, W.; Alexan, N.; Gabr, M. Multiple-layer image encryption utilizing fractional-order chen hyperchaotic map and cryptographically secure prngs. *Fractal Fract.* **2023**, *7*, 287. [\[CrossRef\]](#)
63. Wang, L.; Song, H.; Liu, P. A novel hybrid color image encryption algorithm using two complex chaotic systems. *Opt. Lasers Eng.* **2016**, *77*, 118–125. [\[CrossRef\]](#)
64. Liu, J.; Ma, Y.; Li, S.; Lian, J.; Zhang, X. A new simple chaotic system and its application in medical image encryption. *Multimed. Tools Appl.* **2018**, *77*, 22787–22808. [\[CrossRef\]](#)

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.