frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 03 December 2024
DOI 10.3389/fphy.2024.1494056

OPEN ACCESS

EDITED BY
Hui-Jia Li,
Nankai University, China

REVIEWED BY
Aceng Sambas,
Sultan Zainal Abidin University, Malaysia
Cesar Cruz-Hernandez,
Center for Scientific Research and Higher Education in Ensenada (CICESE), Mexico
Ge Gao,
Beijing Sport University, China

*CORRESPONDENCE
Jianchu Lin,
linjianchu@hyit.edu.cn

RECEIVED 10 September 2024
ACCEPTED 01 November 2024
PUBLISHED 03 December 2024

CITATION
Ge B, Qu G, Shen Z and Lin J (2024) A counter mode and multi-channel based chaotic image encryption algorithm for the internet of things.
Front. Phys. 12:1494056.
doi: 10.3389/fphy.2024.1494056

A counter mode and multi-channel based chaotic image encryption algorithm for the internet of things

Bin Ge[1], Guoqiang Qu[2], Zhihua Shen[1] and Jianchu Lin[2,3]*

[1]Electronic Information Engineering College, Nantong Vocational University, Nantong, China, [2]Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, China, [3]Jiangsu Key Lab of Image and Video Understanding for Social Security, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing, China

To deal with the threat of image privacy leakage in the Internet of things, this paper presents a novel batch images encryption algorithm using the counter mode and a multi-channel processing scheme. We employ multi-thread technique combined with an adapter to construct a novel multi-channel processing scheme, which can encrypt four different sized images in one round. Moreover, the counter encryption mode, which can compute round keys from a plaintext related session key, is introduced to decrease the difficulty of session key management when dealing with batch images. The security tests demonstrate the exceptional performance of the proposed algorithm in terms of security, as evidenced by *P*-values of statistical tests far larger than 0.01, correlation coefficients and entropies of cipher images close to 0 and greater than 7.99. Additionally, the results of NPCR and UACI tests closely approximate the theoretical values 99.6094% and 33.4635%, the proposed algorithm can better resist statistical, exhaustive, differential, or even chosen plaintext attacks. Moreover, due to the novel parallel scheme with a linear time complexity of O(2*W*+2*H*), which demonstrates an acceleration of over 300% compared to existing algorithms, it only takes 2.1sto encrypt one hundred images with varying sizes. Therefore, the proposed algorithm succeeds in exceeding existing algorithms in meeting the efficiency and security requirements for encrypting batch images.

KEYWORDS

internet of things, image encryption, chaotic system, counter encryption mode, multi-thread technique

1 Introduction

With the rapid proliferation of Internet of things (IoT) and wireless communication techniques, multimedia data has become an indispensable facet of daily life. Among various modes of multimedia representation, digital images have gained immense popularity owing to their affordability and easy accessibility. However, a looming threat of privacy breaches casts a shadow over image communication, resulting in financial losses or even endangering lives [1–3]. Scholars have endeavored to address this issue through encryption techniques

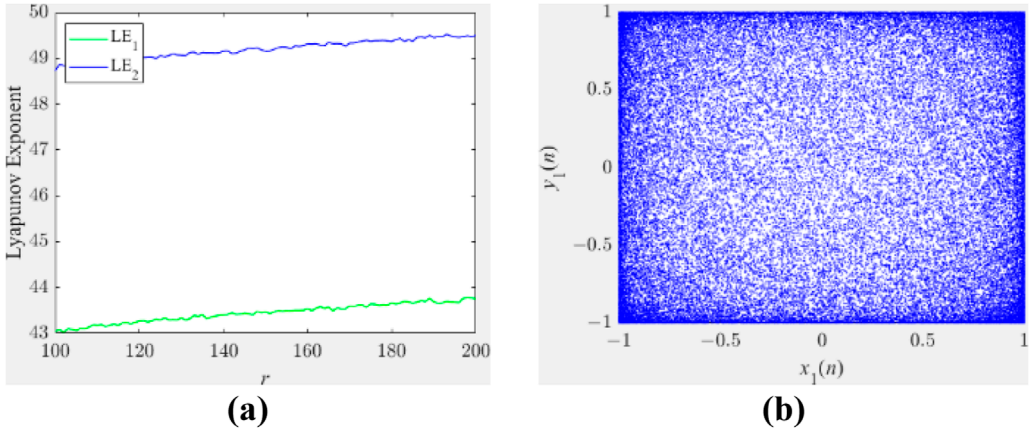

FIGURE 1
The chaotic behavior of 2D-SCCM. **(A)** LE distribution diagram. **(B)** Phase space trajectory diagram.

[4, 5], watermarking [6, 7], steganography [8, 9], among others. Watermarking, whether blind or non-blind, offers creators an effective means to safeguard their copyright for image products. Steganography involves concealing secrets within host images and establishes a channel between sender and receiver for sharing less sensitive private information. Nevertheless, both techniques face challenges when it comes to securely transmitting high-capacity confidential images. Hence, encryption techniques remain crucial in transforming vivid images into disorderly forms as a defense against theft of classified information.

The limitations of well-known commercial ciphers, such as AES and ECC, in encrypting images have led to an increasing focus on the development of a dedicated cryptosystem for image encryption [10, 11]. The readability of a digital image can be compromised through various scrambling methods, such as Zigzag transformation [12, 13], Arnold transformation [14, 15] or magic cube [16, 17]. Therefore, despite the visual appearance of encryption, these methods are still susceptible to attacks based on frequency analysis and other techniques. Therefore, to completely conceal the relationship between the original and cipher images, various diffusion methods have been proposed to convert organized pixels into a state of statistical disorder. In general, a two-round cipher-block-chain (CBC) based diffusion process can provide protection against most cryptographic attacks by incorporating the previous ciphertext to encrypt the current plaintext and achieve an avalanche effect [5, 18, 19]. Additionally, pseudo random numbers play a crucial role in achieving satisfactory diffusion effects. Due to their high initial sensitivity and unpredictable randomness, chaotic systems naturally meet the requirements of an ideal cryptography pseudo random number generator [20–22]. Consequently, chaos-based image encryption algorithms have yielded significant advancements in preserving image privacy. For instance, Hua et al. [23] developed a novel 2D-LSM system to enhance the security of their color image encryption algorithm. Gao et al. [24] utilized DNA mutation operations and hyperchaotic systems to design a secure image encryption algorithm. Huang et al. [25] introduced compressive sensing and DWT techniques for enhanced security against attacks while Teng et al. [26] proposed a novel simultaneous permutation and diffusion structure that offers both high security and fast encryption speed.

However, in an actual communication between IoT devices, it usually involves hundreds of images to transfer. Since most existing algorithms aimed at single image encryption (hereinafter referred to as SIEA, single image encryption algorithm), it might emerge some imperfections when directly applied them to encrypt multiple images. First, in many SIEAs [27–29], the hash function is employed to provide a plaintext related session key which may cause extra communication overhead of session key exchange in multi-image encryption. Second, as the number of images grows, if a SIEA simply encrypts images in serial, the heavy time consumption of encryption will lead to low efficiency of image secure communications [23–29]. Third, some scholars attempted to reuse their SIEAs by composing multiple images into a single image with big size [30–32], but ignored the rapid growth of space complexity along with the increase of image numbers. Hence, researchers have gradually paid more attention to design specific algorithms for multi-image encryption (hereinafter referred to as MIEA, multi-image encryption algorithm).

Inspired by the 3D structure of the color image, some MIEAs operate on a three-dimensional image cube comprising all original images. In conjunction with conventional scrambling and rotation techniques, Sahasrabuddhe et al. [33] successfully achieved three-dimensional image cube scrambling prior to the diffusion process. Zhang et al. [34] applied Sarrus and 3D Fibonacci rules on the image cube to achieve scrambling and diffusion respectively. Zhou et al. [35] proposed a simultaneous scrambling-diffusion method on the image cube. Obviously, suchlike algorithms have an obvious limitation that they can only deal with multiple images with the same size. To address this issue, zero padding is commonly employed as a universal solution for smaller images [33–35]. Furthermore, considering the potential security issues of simple zero padding, Wang et al. [36]

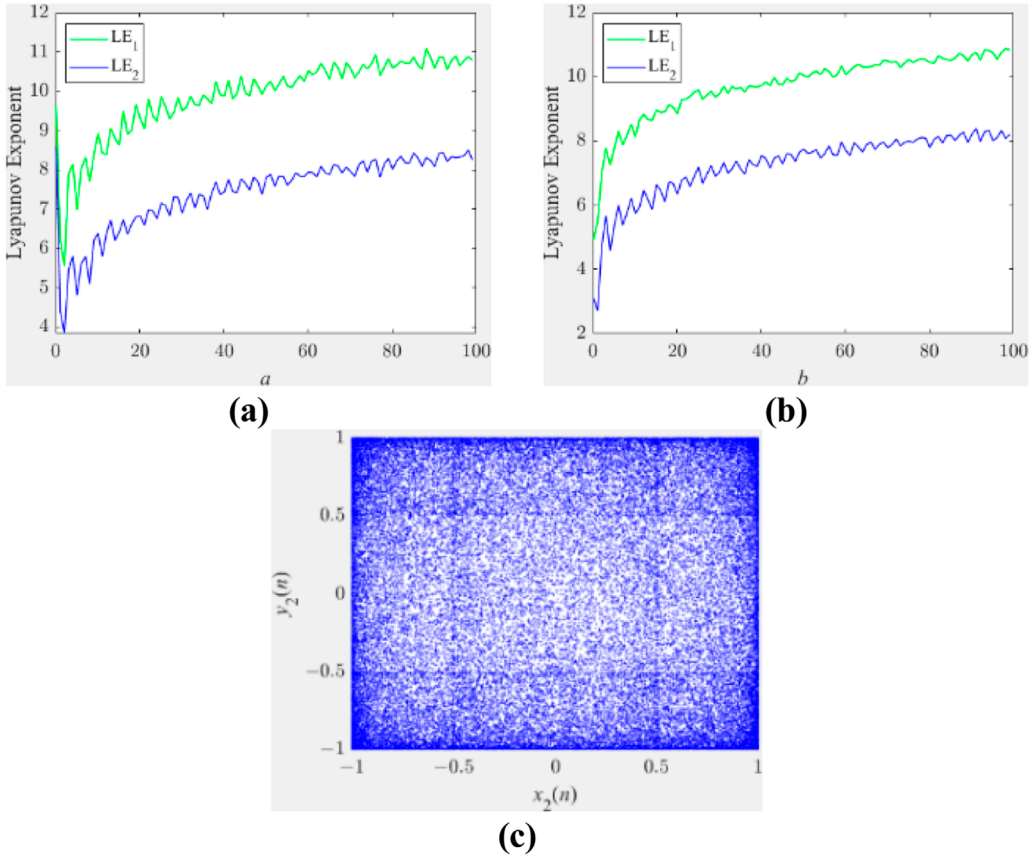

FIGURE 2
The chaotic behavior of 2D-SIDCM. **(A)** LE distribution diagram when b being set as 100. **(B)** LE distribution diagram when a being set as 100. **(C)** Phase space trajectory diagram.

depended on blurred pixels method to fill smaller images to make their algorithm more robust against various attacks. In addition, the implementation of these algorithms necessitates additional computing resources, rendering they impractical for IoT devices with limited resources.

Another universal approach is that existing SIEA can work smoothly on a single image compounded from multiple images. Based on a so called NTMDP method, Tao et al. [37] realized multiple images encryption *via* a base-plane image, but the only shortcoming is that the high complexity of compound process makes their MIEA unsuitable for real-time communications. By using a P-tensor product compressive sensing technique, Xiao et al. [38] also presented a secure MIEA based on a compound single image, which not only had fast encryption speed but also had low storage consumption. However, reconstruction-induced data loss imposes limitations on the application of this method in secure transmissions of high-resolution images. Gao et al. [39] proposed an innovative scheme by integrating multiple images into a single image within the HSV channel. Subsequently, they employed a single-channel based scrambling and diffusion process to obtain the final cipher image. However, the overall efficiency of the MIEA is deemed unsatisfactory due to significant time consumption associated with color channel transformation.

1.1 Motivation and contribution

To design an applicable chaos-based image encryption algorithm for IoT devices, the complexity of the employed chaotic systems is also an important issue that cannot be ignored. The simple structures of low-dimensional chaotic systems make them suitable for implementation in resource-constrained IoT devices [40]. However, they may experience degradation, leading to the transformation of the chaotic system into a periodic system. The high-dimensional, especially hyperchaotic systems exhibit strong robustness against degradation, making them more complex to generate superior random sequences for enhanced security. However, this also poses implementation challenges, particularly in resource-constrained IoT devices [41].

In conclusion, given the escalating demand for secure image communications between IoT devices, the existing MIEAs still have notable deficiencies in terms of both efficiency and security. Therefore, this paper proposes an innovative encryption algorithm for batch images based on counter mode and a multi-channel processing scheme. Utilizing the multi-channel processing scheme, the proposed algorithm enables parallel encryption of four different sized images using any well-established SIEA. By introducing counter mode encryption, the novel scheme can easily

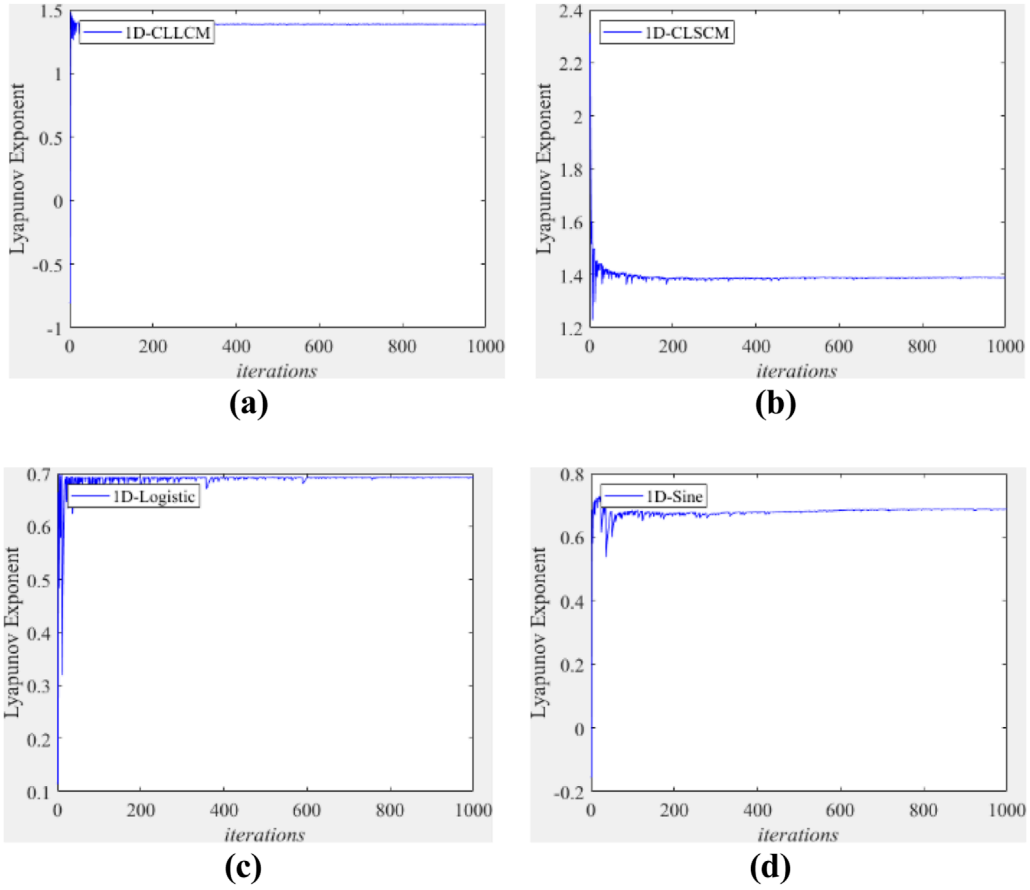

FIGURE 3
Largest Lyapunov exponents of **(A)** 1D-CLLCM, **(B)** 1D-CLSCM, **(C)** 1D-Logistic, and **(D)** 1D-Sine.

be extended to handle batch images without heavy session key management.

The novelties and contributions of the proposed MIEA are summarized below:

1) To protect privacy of image data in IoT device, a fast and secure batch images encryption algorithm based on counter mode and multi-channel processing scheme is proposed.
2) A lightweight multi-channel pseudo random number generator is constructed from two 2D hyperchaotic systems, which can offer exceptional performance even in resource-constrained IoT devices.
3) By designing an adapter for key streams, four different sized images can be parallel encrypted using multi-thread technique, thereby eliminating additional operations on the original images.
4) The counter mode is introduced to obtain round keys from a session key on local device when dealing with batch images, which significantly simplifies session key management between IoT devices.
5) Various kinds of tests are performed on the proposed algorithm to verify its outstanding performances of feasibility, security, and efficiency, which can primely meet the requirements of batch images encryption.

The remainder of this paper is organized as follows. Section 2 presents the preliminaries for this work. In Section 3, we introduce the proposed image encryption algorithm. In Section 4, we present and analyze the experimental results of the proposed algorithm. Finally, we draw conclusions in Section 5.

2 Preliminaries

2.1 The employed 2D hyperchaotic systems

Pseudo random numbers are always critical to achieve high-strength image encryption to resist attacks. The hyperchaotic system, which contains two or more positive Lyapunov exponents, has extremely complex phase space trajectories, strong resistance against degradation in digital systems, and long-term unpredictable randomness. However, considering the heavy time consumption to solve equations, high dimensional continuous hyperchaotic systems are not so proper for time-sensitive encryption occasions. Recently, due to the progress of chaos control theory, 2D hyperchaotic systems have proven to be more suitable for image encryption,

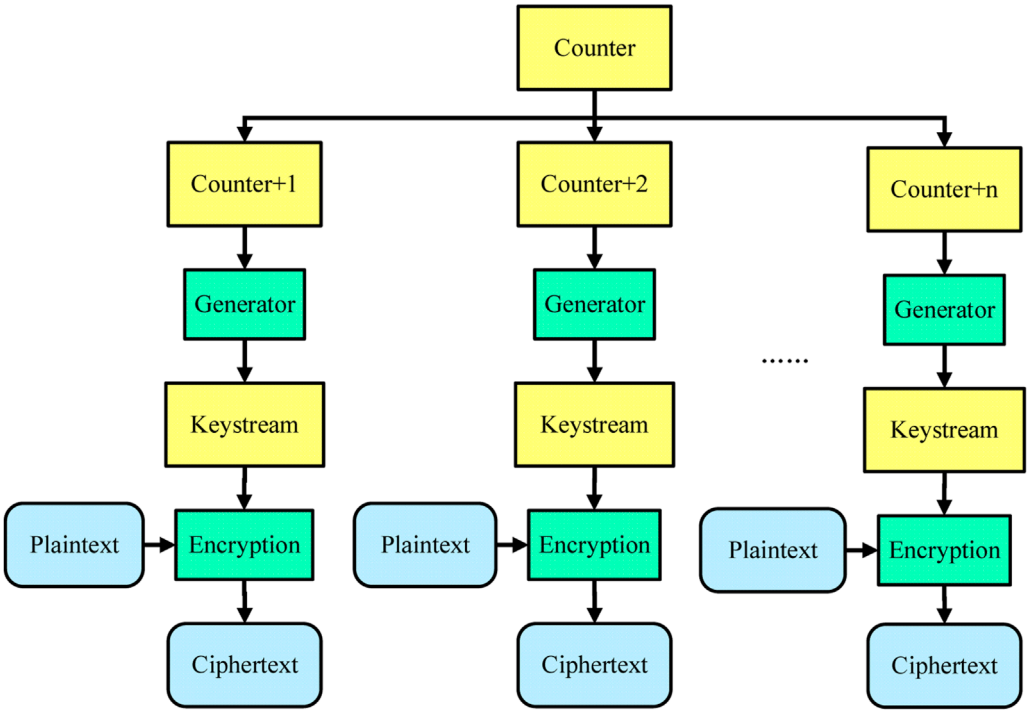

FIGURE 4
The schematic diagram of counter encryption mode.

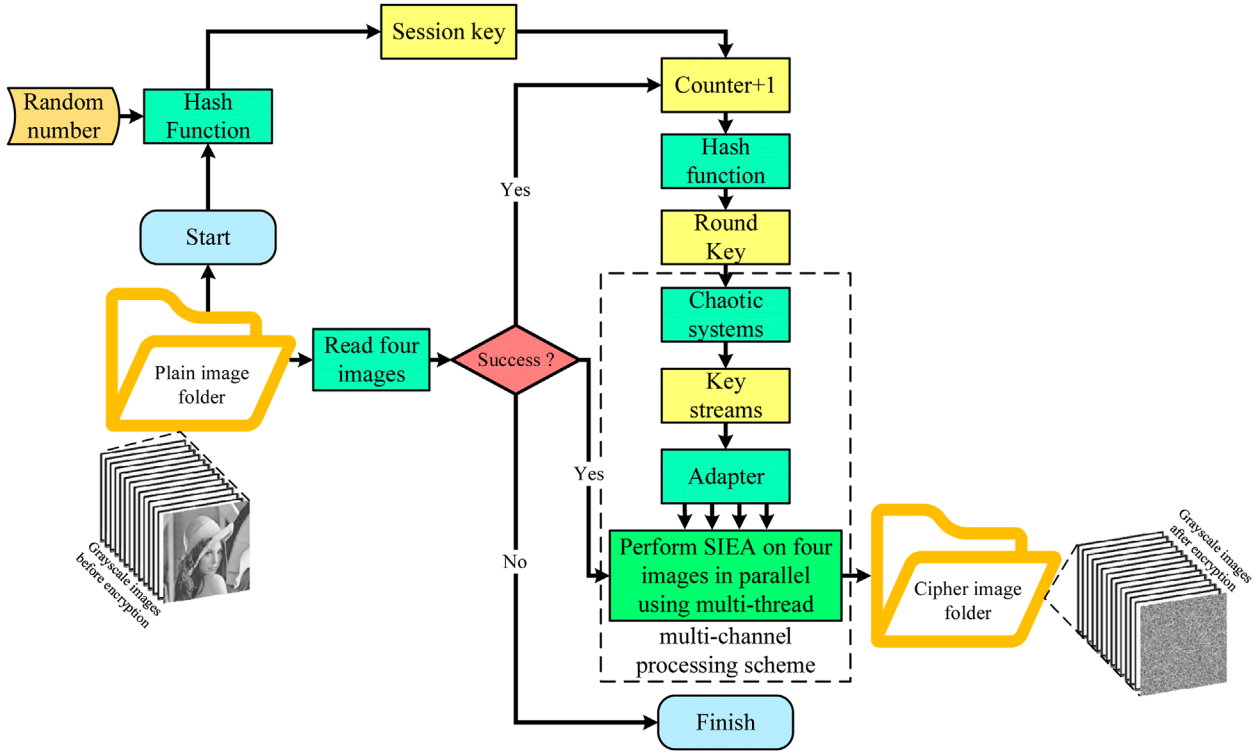

FIGURE 5
The flowchart of the proposed multi-image encryption algorithm.

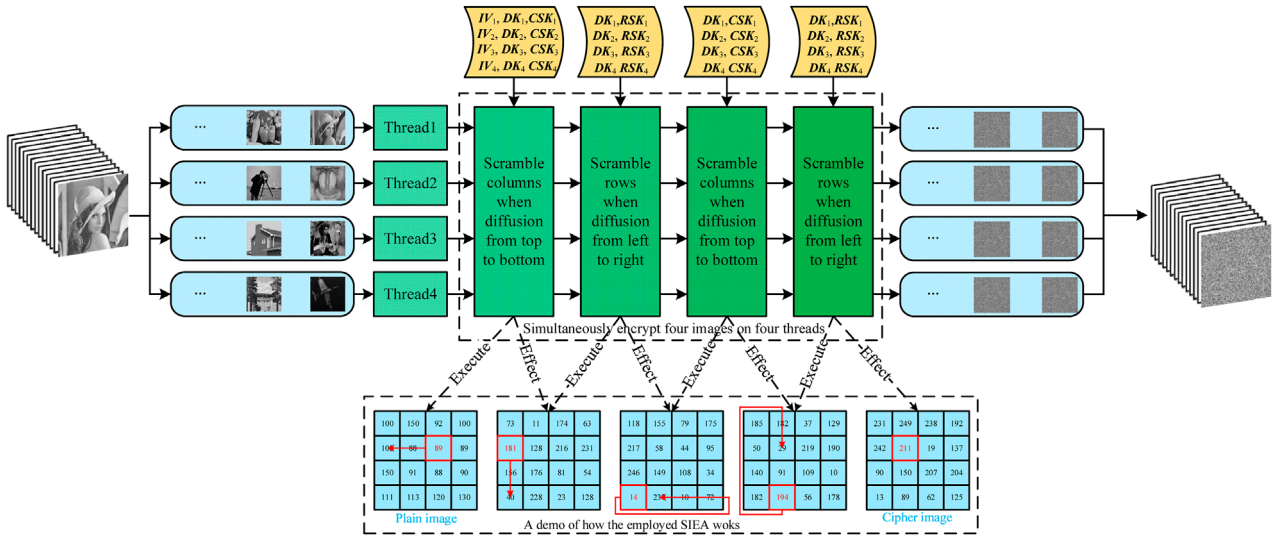

FIGURE 6
The encryption details of the multi-channel processing scheme.

since they have simpler structure and faster iteration speed, while they still maintain good properties of other hyperchaotic systems. In this paper, to provide multi-channel pseudo random numbers with good properties, two novel 2D hyperchaotic systems are employed.

The first 2D hyperchaotic system 2D-SCCM [42] is defined as Equation 1 below:

$$\begin{cases} x_1(n+1) = \sin\left(10^{17} r x_1(n) + y_1(n)\right) \\ y_1(n+1) = \cos\left(10^{19} r \pi y_1(n)\right) \end{cases} \quad (1)$$

where r is the control parameter, and x_1, $y_1 \in [-1,0) \bigcup (0,1]$. When r locates in 100 and 200, as Figure 1 shows, two positive Lyapunov exponents occur, indicating the 2D-SCCM evolves into hyperchaotic state. In this paper, for better encryption effect, r is set as 200.

The another 2D hyperchaotic system 2D-SIDCM [43] is described as Equation 2:

$$\begin{cases} x_2(n+1) = \sin\left(\dfrac{b}{\sin\left(a y_2(n)\right)}\right) \\ y_2(n+1) = \sin\left(a \sin\left(\dfrac{b}{x_2(n) y_2(n)}\right)\right) \end{cases} \quad (2)$$

which has two control parameters a, b and x_2, $y_2 \in [-1,0) \bigcup (0,1]$. When $a, b \in (0, 100]$, it contains two positive Lyapunov exponents and represents hyperchaotic behaviors as demonstrated in Figure 2. In this paper, for high-strength encryption, a and b are both set as 100.

2.2 The employed 1D chaotic maps

The 1D chaotic system is a proper tool for fast scrambling pixel positions. However, traditional 1D chaotic systems, such as Logistic map and Sine map, suffer from degeneration. Recently, by introducing cascade technique, many more robust 1D chaotic systems are proposed [44].

On the one hand, for scrambling, a cascade chaotic map 1D-CLLCM is employed, which is given by Equation 3, and $x_3 \in [-1,0) \bigcup (0,1]$.

$$x_3(n+1) = 1 - 2\left(1 - 2x_3(n)^2\right)^2 \quad (3)$$

On the other hand, for providing initial vectors of the encryption process, another cascade chaotic map 1D-CLSCM, as shown in Equation 4 with $x_4 \in (0,1]$, is utilized.

$$x_4(n+1) = 4 \sin\left(\pi x_4(n)\right)\left(1 - \sin\left(\pi x_4(n)\right)\right) \quad (4)$$

By comparing between Figures 3A, B and Figures 3C, D, two cascade chaotic maps have much larger Lyapunov exponents than traditional maps, meaning better chaotic properties against degeneration. Thus, they can ensure better scrambling effect and provide initial vectors with enough randomness.

2.3 The counter encryption mode

The counter mode was first introduced to turn a block cipher into a stream cipher [45, 46]. As demonstrated in Figure 4, the counter mode generates next keystream by encrypting successive counter values.

In this paper, the counter mode is employed to generate round key of encrypting each four original images, which can reduce the difficulty of session key management. And as we can see in Figure 4, if all counter values are obtained before encryption, each round encryption can be parallel processed, which can further accelerate batch images encryption.

TABLE 1 The employed single image encryption algorithm.

Algorithm SSDIEA
Input: ***DK***, ***CSK***, ***RSK***, ***IV***, ***P*** (with width W and height H)
Output: ***C***
1: Create an empty matrix ***TD*** which stores the temporary data during encryption
/*Perform columns scrambling when diffusion from top to bottom using ***IV***, ***DK***, and ***CSK***/
2: $\boldsymbol{TD}(1,:) \leftarrow (\boldsymbol{IV} + \boldsymbol{P}(1,:))\%256$
3: $\boldsymbol{TD}(1,:) \leftarrow \boldsymbol{TD}(1,:) \oplus \boldsymbol{DK}(1,:)$
4: $\boldsymbol{TD}(1,:) \leftarrow \boldsymbol{TD}(1,:) << \boldsymbol{CSK}(1))$// operator << represents a circular shift function
/*Continue to deal with the remaining rows*/
5: **for** i from 2 to H
6: $\boldsymbol{TD}(i,:) \leftarrow (\boldsymbol{TD}(i\text{-}1,:) + \boldsymbol{P}(i,:))\%256$
7: $\boldsymbol{TD}(i,:) \leftarrow \boldsymbol{TD}(i,:) \oplus \boldsymbol{DK}(i,:)$
8: $\boldsymbol{TD}(i,:) \leftarrow \boldsymbol{TD}(i,:) << \boldsymbol{CSK}(i))$
9: **end for**
/*Perform rows scrambling when diffusion from left to right using ***DK*** and ***RSK***, and let the last column of the previous encryption phase be the new initial vector*/
10: $\boldsymbol{TD}(:,1) \leftarrow (\boldsymbol{TD}(:,W) + \boldsymbol{TD}(:,1))\%256$
11: $\boldsymbol{TD}(:,1) \leftarrow \boldsymbol{TD}(:,1) \oplus \boldsymbol{DK}(:,1)$
12: $\boldsymbol{TD}(:,1) \leftarrow \boldsymbol{TD}(:,1) << \boldsymbol{RSK}(1))$
/*Continue to deal with the remaining columns*/
13: **for** i from 2 to W
14: $\boldsymbol{TD}(:,i) \leftarrow (\boldsymbol{TD}(:,i\text{-}1) + \boldsymbol{TD}(:,i))\%256$
15: $\boldsymbol{TD}(:,i) \leftarrow \boldsymbol{TD}(:,i) \oplus \boldsymbol{DK}(:,i)$
16: $\boldsymbol{TD}(:,i) \leftarrow \boldsymbol{TD}(:,i) << \boldsymbol{RSK}(i))$
17: **end for**
/*Perform columns scrambling when diffusion from top to bottom using ***DK*** and ***CSK***, and let the last row of the previous encryption phase be the new initial vector*/
18: $\boldsymbol{TD}(1,:) \leftarrow (\boldsymbol{TD}(H,:) + \boldsymbol{TD}(1,:))\%256$
19: $\boldsymbol{TD}(1,:) \leftarrow \boldsymbol{TD}(1,:) \oplus \boldsymbol{DK}(1,:)$
20: $\boldsymbol{TD}(1,:) \leftarrow \boldsymbol{TD}(1,:) << \boldsymbol{CSK}(1))$//
/*Continue to deal with the remaining rows*/
21: **for** i from 2 to H

(Continued on the following page)

TABLE 1 (*Continued*) The employed single image encryption algorithm.

Algorithm SSDIEA
22: $\boldsymbol{TD}(i,:) \leftarrow (\boldsymbol{TD}(i\text{-}1,:) + \boldsymbol{TD}(i,:))\%256$
23: $\boldsymbol{TD}(i,:) \leftarrow \boldsymbol{TD}(i,:) \oplus \boldsymbol{DK}(i,:)$
24: $\boldsymbol{TD}(i,:) \leftarrow \boldsymbol{TD}(i,:) << \boldsymbol{CSK}(i))$
25: **end for**
/*Perform rows scrambling when diffusion from left to right using ***DK*** and ***RSK***, and let the last column of the previous encryption phase be the new initial vector*/
26: $\boldsymbol{C}(:,1) \leftarrow (\boldsymbol{TD}(:,W) + \boldsymbol{TD}(:,1))\%256$
27: $\boldsymbol{C}(:,1) \leftarrow \boldsymbol{TD}(:,1) \oplus \boldsymbol{DK}(:,1)$
28: $\boldsymbol{C}(:,1) \leftarrow \boldsymbol{TD}(:,1) << \boldsymbol{RSK}(1))$
/*Continue to deal with the remaining columns*/
29: **for** i from 2 to W
30: $\boldsymbol{C}(:,i) \leftarrow (\boldsymbol{TD}(:,i\text{-}1) + \boldsymbol{TD}(:,i))\%256$
31: $\boldsymbol{C}(:,i) \leftarrow \boldsymbol{TD}(:,i) \oplus \boldsymbol{DK}(:,i)$
32: $\boldsymbol{C}(:,i) \leftarrow \boldsymbol{TD}(:,i) << \boldsymbol{RSK}(i))$
33: **end for**
34: Delete ***TD***

3 The proposed batch images encryption algorithm

The flowchart of the proposed algorithm is demonstrated in Figure 5. Then, the details will be discussed from 1) how to obtain session key and round key, 2) how to generate the required key streams, 3) the employed SIEA, and 4) how the proposed algorithm works.

3.1 The management of session key and round keys

To simplify session key management for batch images encryption task, this paper employs both hash function and random number. First, to resist differential attacks, the hash function is employed to quickly extract a global session key being sensitive to all under encrypted images. Second, to avoid invalid encryption when facing session key leakage, an external true random number is introduced to achieve one-time pad. At last, to accomplish encryption task, round keys for each four images will be calculated from the global session key.

The detail steps of generating session key and round keys are described below:

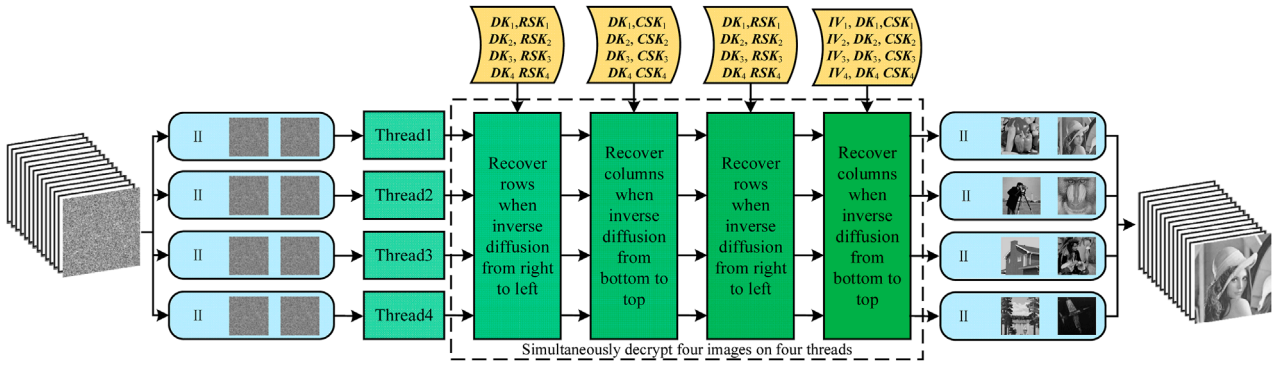

FIGURE 7
The decryption details of the multi-channel processing scheme.

Step A1: Input an external 256-bit random number K_e and all original images into a secure hash function SHA-256, then a 256-bit session key SK can be created.

Step A2: For encrypting each four images, the SK will be further calculated by SHA-256 to obtain round keys, and since the number of images N is certain for every encryption task, round keys RK_1, RK_2, … , $RK_{N/4}$ can be obtained far before encryption.

Step A3: To launch chaotic systems, the round key will be post processed into floating point numbers. Take the round key RK_1 as an example, it can be converted into initial values for 2D-SCCM, 2D-SIDCM, 1D-CLLCM, and 1D-CLSCM by Equation 5, where operator ⊕ means a bitwise exclusive or operation:

$$\begin{cases} x_1(1) = (\boldsymbol{RK}_1(1{:}40) \oplus \boldsymbol{RK}_1(217{:}256))/2^{40} \\ y_1(1) = (\boldsymbol{RK}_1(41{:}80) \oplus \boldsymbol{RK}_1(177{:}216))/2^{40} \\ x_2(1) = (\boldsymbol{RK}_1(81{:}120) \oplus \boldsymbol{RK}_1(137{:}176))/2^{40} \\ y_2(1) = (\boldsymbol{RK}_1(121{:}160) \oplus \boldsymbol{RK}_1(97{:}136))/2^{40} \\ x_3(1) = (\boldsymbol{RK}_1(161{:}200) \oplus \boldsymbol{RK}_1(57{:}96))/2^{40} \\ x_4(1) = (\boldsymbol{RK}_1(201{:}240) \oplus \boldsymbol{RK}_1(17{:}56))/2^{40} \end{cases} \tag{5}$$

3.2 The generation of key streams for diffusion

To perform multi-thread encryption on each four images, we combine 2D-SCCM and 2D-SIDCM to construct a multi-channel pseudo random numbers generator for diffusion, which not only has long-term unpredictable randomness, but also has ultra-fast generation speed. It consists of the following steps:

Step B1: Input $x_1(1)$, $y_1(1)$ and $x_2(1)$, $y_2(1)$ into 2D-SCCM and 2D-SIDCM respectively.

Step B2: Pre-iterate 2D-SCCM and 2D-SIDCM 100 times to overcome transient effect for better security, and let new $x_1(1)$, $y_1(1)$ and $x_2(1)$, $y_2(1)$ be the last state values.

Step B3: Create four empty sequences S_1, S_2, S_3, S_4.

Step B4: Read four images and obtain their width W_1, W_2, W_3, W_4, and height H_1, H_2, H_3, H_4.

Step B5: Let $W_{max} = \max\{W_1, W_2, W_3, W_4\}$ and $H_{max} = \max\{H_1, H_2, H_3, H_4\}$.

Step B6: Continue to iterate 2D-SCCM and 2D-SIDCM $W_{max} \times H_{max}$ times and fill S_1, S_2, S_3, S_4 by Equation 6.

$$\begin{cases} \boldsymbol{S}_1 = \{\boldsymbol{S}_1, x_1(i)\}, \boldsymbol{S}_2 = \{\boldsymbol{S}_2, y_1(i)\} \\ \boldsymbol{S}_3 = \{\boldsymbol{S}_3, x_2(i)\}, \boldsymbol{S}_4 = \{\boldsymbol{S}_4, y_2(i)\} \\ i = 1, 2, \cdots, W_{max} \times H_{max} - 1, W_{max} \times H_{max} \end{cases} \tag{6}$$

Step B7: Convert each element of the above generated sequences into integer number which locates in 0–255 as Equation 7 presents.

$$\begin{cases} \boldsymbol{S}_1 = \lfloor (|\boldsymbol{S}_1| - \lfloor |\boldsymbol{S}_1| \rfloor) \times 10^{15} \rfloor \% 256 \\ \boldsymbol{S}_2 = \lfloor (|\boldsymbol{S}_2| - \lfloor |\boldsymbol{S}_2| \rfloor) \times 10^{15} \rfloor \% 256 \\ \boldsymbol{S}_3 = \lfloor (|\boldsymbol{S}_3| - \lfloor |\boldsymbol{S}_3| \rfloor) \times 10^{15} \rfloor \% 256 \\ \boldsymbol{S}_4 = \lfloor (|\boldsymbol{S}_4| - \lfloor |\boldsymbol{S}_4| \rfloor) \times 10^{15} \rfloor \% 256 \end{cases} \tag{7}$$

Step B8: Create four empty sequences DK_1, DK_2, DK_3, DK_4, then fill them by Equation 8.

$$\begin{cases} \boldsymbol{DK}_1 = \boldsymbol{S}_1 \oplus \boldsymbol{S}_3 \\ \boldsymbol{DK}_2 = \boldsymbol{S}_2 \oplus \boldsymbol{S}_4 \\ \boldsymbol{DK}_3 = \boldsymbol{S}_1 \oplus \boldsymbol{S}_4 \\ \boldsymbol{DK}_4 = \boldsymbol{S}_2 \oplus \boldsymbol{S}_3 \end{cases} \tag{8}$$

Step B9: Delete S_1, S_2, S_3, S_4, then reshape DK_1, DK_2, DK_3, DK_4 to four matrices with size of $W_{max} \times H_{max}$.

Step B10: At last, adapt the sizes of DK_1, DK_2, DK_3, DK_4 to sizes of $W_1 \times H_1$, $W_2 \times H_2$, $W_3 \times H_3$, $W_4 \times H_4$, then obtain four diffusion key matrices for four different size images.

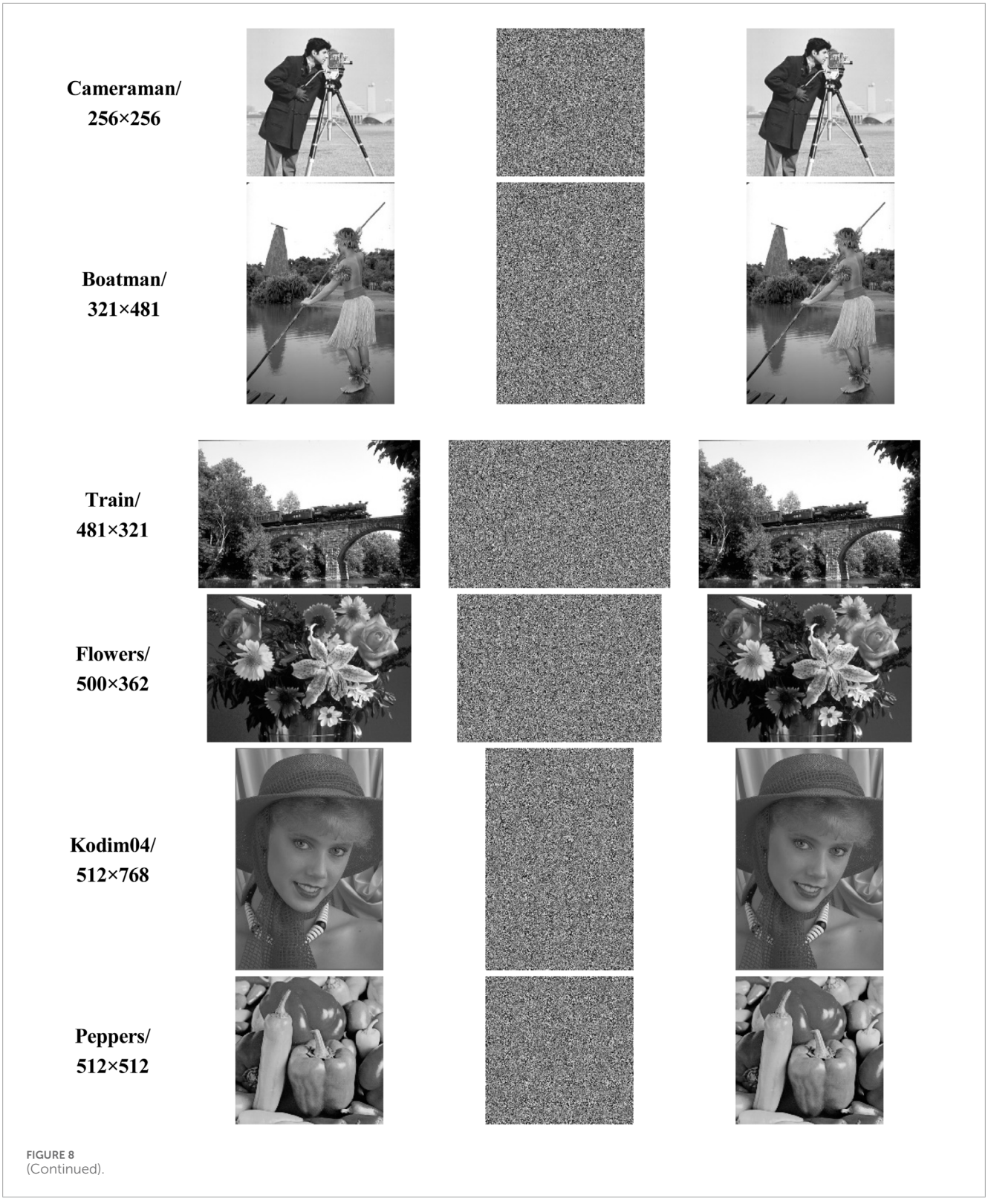

FIGURE 8
(Continued).

3.3 The generation of key streams for initial vectors

A random initial vector is prerequisite to perform a favorable CBC-based diffusion process. Thus, we utilize the 1D-CLLCM to provide enough pseudo random numbers with good randomness and unpredictability as an initial vector.

Step C1: Input $x_3(1)$ into 1D-CLLCM.

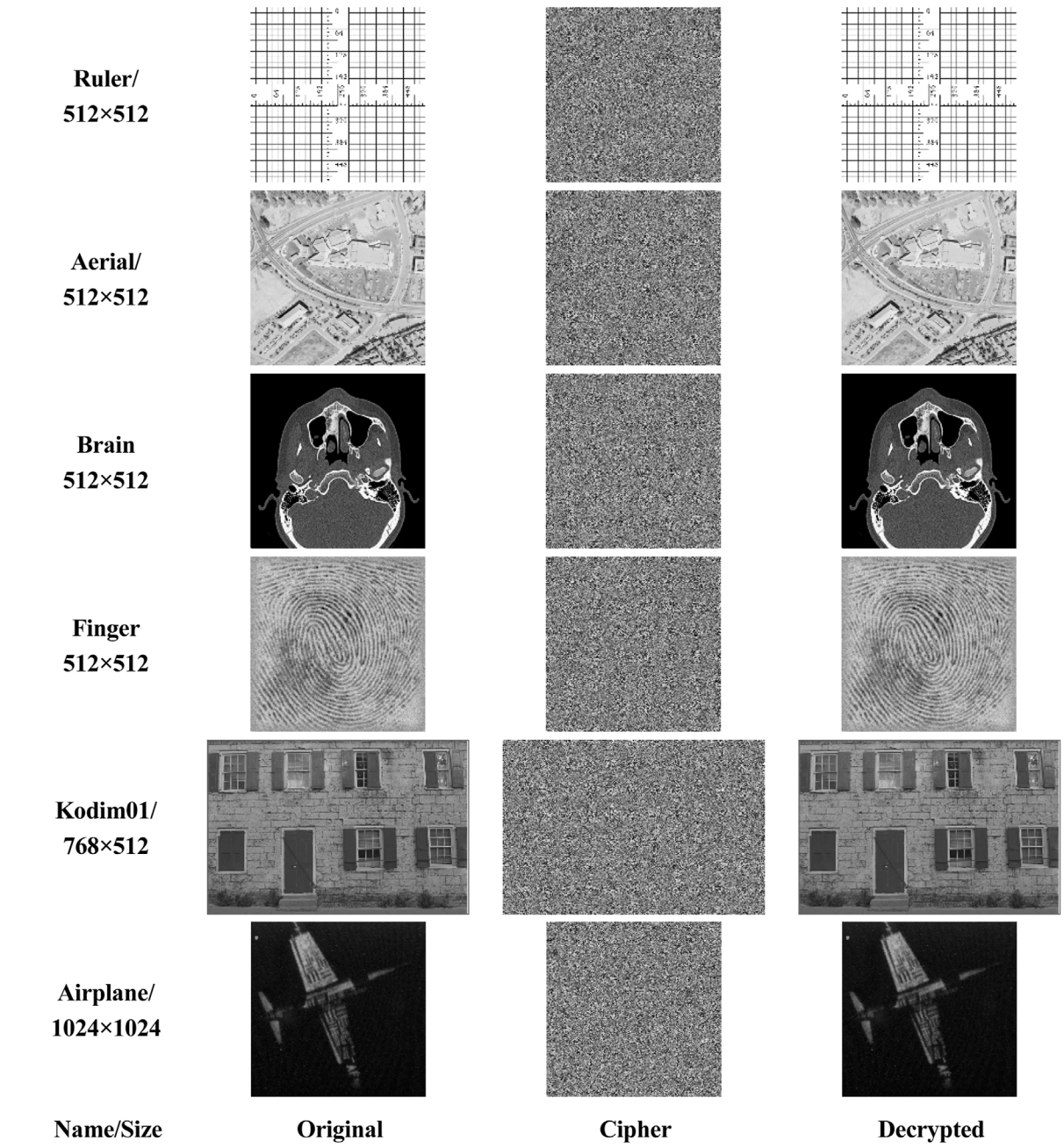

FIGURE 8
(Continued). Simulation results. The original images, the cipher images, and the decrypted images in second to fourth columns respectively, and their sizes are listed in the first column.

Step C2: Pre-iterate 1D-CLLCM 100 times to overcome transient effect for better security, and let new $x_3(1)$ be the last state value.

Step C3: Create an empty sequences S.

Step C4: Continue to iterate 1D-CLLCM $W_{\max}$ times and let $S = \{x_3(i), i = 1, 2, \ldots, W_{\max}\text{-}1, W_{\max}\}$.

Step C5: Convert all elements of S into integer numbers locating in 0–255 by Equation 9.

$$\boldsymbol{S} = \lfloor |\boldsymbol{S}| \times 10^{15} \rfloor \% 256 \tag{9}$$

Step C6: Delete S, then obtain the four initial vectors IV_1, IV_2, IV_3, IV_4, by Equation 10.

$$\begin{cases} \boldsymbol{IV}_1 = \boldsymbol{S}(W_{\max} - W_1 + 1 : W_{\max}) \\ \boldsymbol{IV}_2 = \boldsymbol{S}(W_{\max} - W_2 + 1 : W_{\max}) \\ \boldsymbol{IV}_3 = \boldsymbol{S}(W_{\max} - W_3 + 1 : W_{\max}) \\ \boldsymbol{IV}_4 = \boldsymbol{S}(W_{\max} - W_4 + 1 : W_{\max}) \end{cases} \tag{10}$$

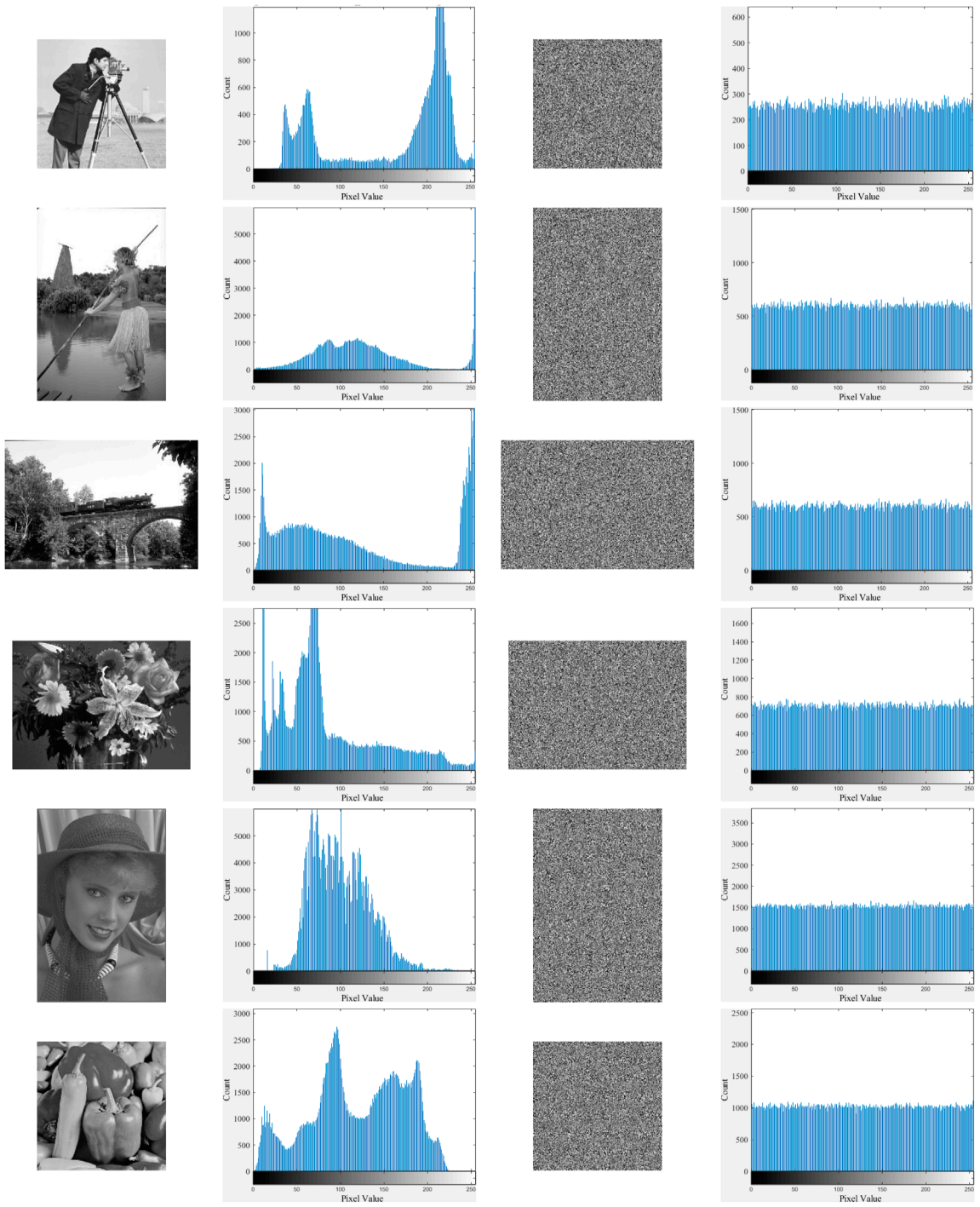

FIGURE 9
(Continued).

3.4 The generation of key streams for scrambling

To further complex the encryption process, we utilize 1D-CLSCM to generate key stream for scrambling by the following steps:

Step D1: Input $x_4(1)$ into 1D-CLSCM.

Step D2: Pre-iterate 1D-CLSCM 100 times to overcome transient effect for better security, and let new $x_4(1)$ be the last state value.

Step D3: Create an empty sequences S.

Step D4: Continue to iterate 1D-CLLCM W_{max} times and fill $S = \{x_4(i), i = 1, 2, \ldots, W_{max}\text{-}1, W_{max}\}$

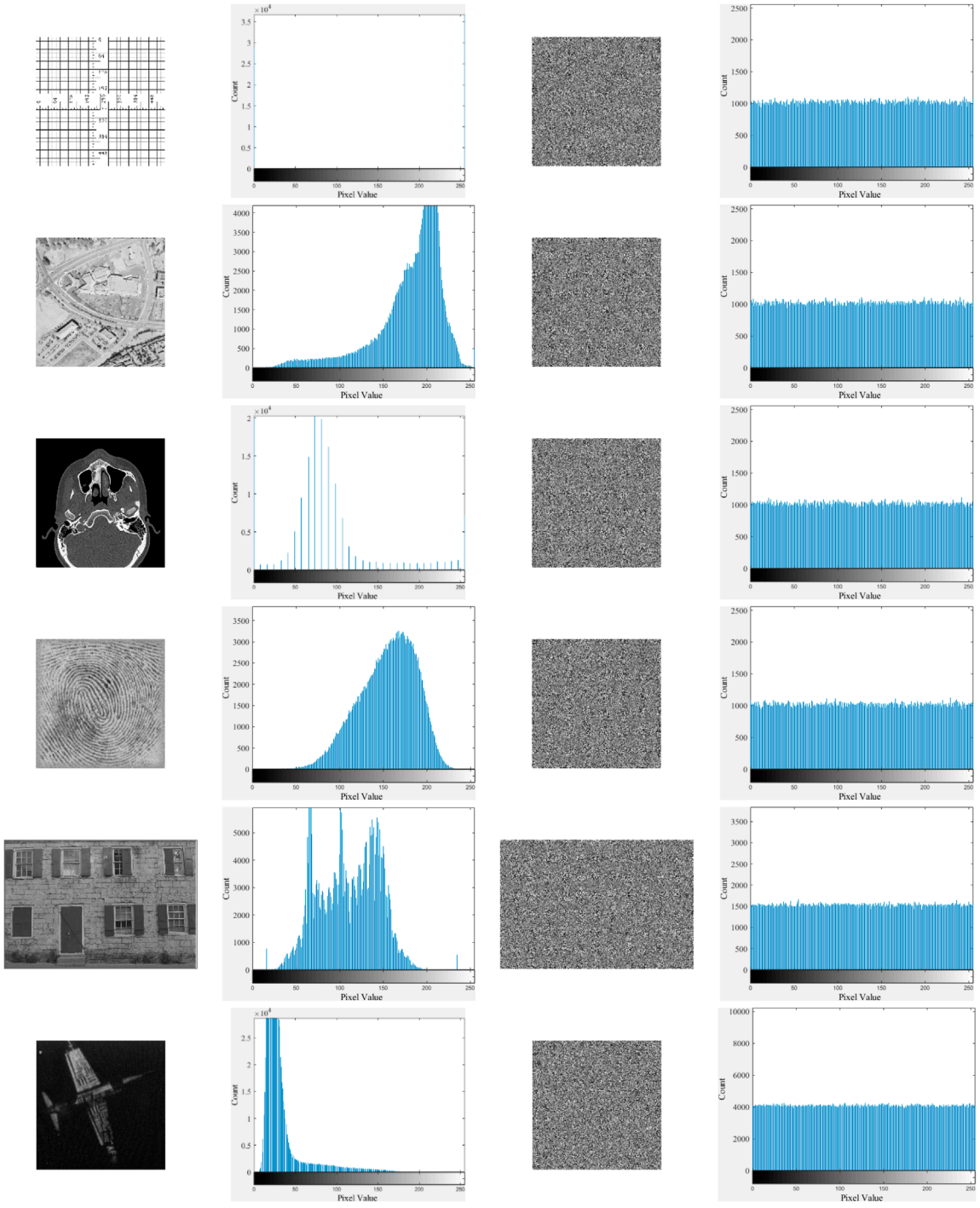

FIGURE 9
(Continued). Results of histogram test. The original images, the cipher images, and the decrypted images in second to fourth columns respectively, and their sizes are listed in the first column. Moreover, the chi-square ($\chi 2$) test [as defined in Equation 15] is utilized to further assess the uniformity of histograms in a more rigorous manner.

TABLE 2 Results of χ^2 test.

Image	Original	Cipher	*P*-value	Result
Cameraman	1.6127×10^5	265.7500	0.8739	Pass
Boatman	2.2739×10^6	276.7912	0.7470	Pass
Train	4.6733×10^5	269.6517	0.8350	Pass
Flowers	2.5826×10^5	260.4945	0.9160	Pass
Kodim04	5.6490×10^5	257.9727	0.9321	Pass
Peppers	5.4023×10^5	289.1185	0.9999	Pass
Ruler	5.3703×10^7	240.6309	0.9890	Pass
Aerial	4.4186×10^5	241.9336	0.9871	Pass
Brain	1.6218×10^7	243.6484	0.9842	Pass
Finger	3.2698×10^5	259.2285	0.9244	Pass
Kodim01	1.2017×10^5	225.9395	0.9986	Pass
Airplane	7.1999×10^5	245.4497	0.9805	Pass

Step D5: Create four empty sequences S_1, S_2, S_3, S_4, then fill them by Equation 11.

$$\begin{cases} \boldsymbol{S}_1 = \boldsymbol{S}(W_{\max} - W_1 + 1{:}W_{\max}) \\ \boldsymbol{S}_2 = \boldsymbol{S}(W_{\max} - W_2 + 1{:}W_{\max}) \\ \boldsymbol{S}_3 = \boldsymbol{S}(W_{\max} - W_3 + 1{:}W_{\max}) \\ \boldsymbol{S}_4 = \boldsymbol{S}(W_{\max} - W_4 + 1{:}W_{\max}) \end{cases} \tag{11}$$

Step D6: Obtain CSK_1, CSK_2, CSK_3, CSK_4 for scrambling columns of four images by Equation 12.

$$\begin{cases} \boldsymbol{CSK}_1 = \lfloor (\boldsymbol{S}_1 - \lfloor \boldsymbol{S}_1 \rfloor) \times 10^{15} \rfloor \% H_1 \\ \boldsymbol{CSK}_2 = \lfloor (\boldsymbol{S}_2 - \lfloor \boldsymbol{S}_2 \rfloor) \times 10^{15} \rfloor \% H_2 \\ \boldsymbol{CSK}_3 = \lfloor (\boldsymbol{S}_3 - \lfloor \boldsymbol{S}_3 \rfloor) \times 10^{15} \rfloor \% H_3 \\ \boldsymbol{CSK}_4 = \lfloor (\boldsymbol{S}_4 - \lfloor \boldsymbol{S}_4 \rfloor) \times 10^{15} \rfloor \% H_4 \end{cases} \tag{12}$$

Step D7: Empty S, S_1, S_2, S_3, S_4, then continue to iterate1D-CLSCM 100 times, and let new $x_4(1)$ be the last state value.

Step D8: Continue to iterate 1D-CLLCM $H_{\max}$ times and fill $S = \{x_4(i), i = 1, 2, \ldots, H_{\max}\text{-}1, H_{\max}\}$

Step D9: Fill S_1, S_2, S_3, S_4 by Equation 13.

$$\begin{cases} \boldsymbol{S}_1 = \boldsymbol{S}(H_{\max} - H_1 + 1{:}H_{\max}) \\ \boldsymbol{S}_2 = \boldsymbol{S}(H_{\max} - H_2 + 1{:}H_{\max}) \\ \boldsymbol{S}_3 = \boldsymbol{S}(H_{\max} - H_3 + 1{:}H_{\max}) \\ \boldsymbol{S}_4 = \boldsymbol{S}(H_{\max} - H_4 + 1{:}H_{\max}) \end{cases} \tag{13}$$

Step D10: Obtain RSK_1, RSK_2, RSK_3, RSK_4 for scrambling rows of four images by Equation 14.

$$\begin{cases} \boldsymbol{RSK}_1 = \lfloor (\boldsymbol{S}_1 - \lfloor \boldsymbol{S}_1 \rfloor) \times 10^{15} \rfloor \% W_1 \\ \boldsymbol{RSK}_2 = \lfloor (\boldsymbol{S}_2 - \lfloor \boldsymbol{S}_2 \rfloor) \times 10^{15} \rfloor \% W_2 \\ \boldsymbol{RSK}_3 = \lfloor (\boldsymbol{S}_3 - \lfloor \boldsymbol{S}_3 \rfloor) \times 10^{15} \rfloor \% W_3 \\ \boldsymbol{RSK}_4 = \lfloor (\boldsymbol{S}_4 - \lfloor \boldsymbol{S}_4 \rfloor) \times 10^{15} \rfloor \% W_4 \end{cases} \tag{14}$$

Step D11: At last, delete S, S_1, S_2, S_3, S_4.

3.5 The encryption process

In this paper, we depend on the multi-thread technique to parallel encrypt four images in one round as shown in Figure 6.

On each thread, a simultaneous scrambling and diffusion image encryption algorithm (as shown in Table 1: Algorithm SSDIEA) deal with original images one by one.

Then, the encryption process of the proposed algorithm is detailed below:

Step E1: Open four threads.

Step E2: Input four images into four threads respectively.

Step E3: Use round key RK_i, and perform *Step B1-B10*, *Step C1-C6*, *Step D1-D10* to obtain key streams for encryption.

Step E4: Input key streams into each thread respectively as shown in Figure 6, then perform Algorithm SSDIEA to encrypt four images in parallel.

Step E5: Repeat *Step E2-E4* until all images being encrypted.

Step E6: Close all threads.

3.6 The decryption process

Since an 8-bit pixel value locates in [0,255], all operations of Algorithm SSDIEA have their inverse operations to recover the original image from a cipher image. Then the decryption process of the proposed batch images encryption algorithm is illustrated in Figure 7.

4 Experimental results and analyses

In this section, the proposed algorithm is assessed from both security and efficiency aspects. All the employed tools and the proposed algorithm are implemented on MATLAB R2021b platform using a workstation equipped with 12th Gen Intel(R) Core(TM)

TABLE 3 Results of randomness tests (set α = 0.01).

Test	*P*-value													
	Key stream for		Cameraman	Boatman	Train	Flowers	Kodim04	Peppers	Ruler	Aerial	Brain	Finger	Kodim01	Airplane
	Scrambling	Diffusion												
Frequency	0.5831	0.5509	0.8142	0.6656	0.1604	0.6080	0.7648	0.8173	0.9072	0.9423	0.9878	0.6391	0.4738	0.5894
Block Frequency	0.3520	0.2085	0.1784	0.2759	0.4291	0.5608	0.6233	0.7573	0.2719	0.3334	0.6868	0.3666	0.3991	0.3947
Cumulative Sums	0.6225	0.6454	0.3705	0.4266	0.8833	0.2551	0.7672	0.5012	0.2574	0.4607	0.3049	0.8863	0.7369	0.6837
Runs	0.1528	0.6225	0.3157	0.1217	0.5493	0.4761	0.4970	0.2766	0.3659	0.225	0.5703	0.6548	0.4792	0.4216
Longest Runs of Ones	0.8165	0.2353	0.9456	0.1968	0.8199	0.5304	0.2236	0.3217	0.85	0.3344	0.7602	0.4226	0.6536	0.9376
Rank	0.2926	0.3136	0.6813	0.2619	0.5267	0.3499	0.6636	0.2882	0.7502	0.2109	0.8786	0.9124	0.5901	0.1259
Spectral DFT	0.3306	0.1711	0.8419	0.1534	0.7499	0.5064	0.7057	0.4547	0.819	0.2316	0.716	0.1741	0.9711	0.7377
Nonperiodic Template Matchings	0.6386	0.5533	0.5689	0.4449	0.6037	0.8446	0.1077	0.8438	0.7532	0.8557	0.9038	0.29	0.4338	0.5909
Overlapping Template Matchings	0.2208	0.6349	0.4036	0.9284	0.1246	0.2201	0.4187	0.5341	0.4898	0.7131	0.8772	0.9983	0.847	0.4132
Universal Statistical	0.9299	0.3611	0.4715	0.4917	0.3178	0.2979	0.2379	0.4313	0.1911	0.9545	0.6658	0.4411	0.3616	0.6863
Approximate Entropy	0.4065	0.6887	0.7546	0.7299	0.8412	0.1353	0.6433	0.4412	0.7772	0.5309	0.7037	0.7248	0.4242	0.8054
Random Excursions	0.5236	0.4856	0.5609	0.6469	0.1923	0.2084	0.2441	0.8797	0.7183	0.4956	0.1943	0.8619	0.7586	0.8474
Random Excursions Variant	0.7532	0.6364	0.9658	0.5189	0.2109	0.2145	0.2853	0.4411	0.8952	0.4715	0.7971	0.8757	0.6597	0.9837
Serial	0.9764	0.6315	0.8864	0.2063	0.3418	0.8377	0.2824	0.8861	0.2693	0.3888	0.7799	0.2691	0.3215	0.2611
Linear Complexity	0.6724	0.4549	0.1081	0.5166	0.6761	0.2184	0.6336	0.3672	0.8128	0.556	0.3905	0.6562	0.3822	0.6788
Result	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

10.3389/fphy.2024.1494056

TABLE 4 Results of information entropy test.

Image	Original	Cipher
Cameraman	6.9046	7.9971
Boatman	6.4118	7.9987
Train	7.1534	7.9987
Flowers	7.3037	7.9990
Kodim04	6.9467	7.9995
Peppers	6.9691	7.9997
Ruler	0.5000	7.9993
Aerial	6.9940	7.9996
Brain	3.1047	7.9995
Finger	7.0153	7.9997
Kodim01	7.5937	7.9994
Airplane	5.6415	7.9998

TABLE 5 Comparisons with other works.

Algorithm	Cameraman	Peppers
Ours	7.9971	7.9997
Reference [34]	—	7.9997
Reference [35]	7.9999	—
Reference [36]	—	7.9992
Reference [37]	7.9917	7.9917
Reference [39]	7.9972	7.9960

i7-12700H 2.30 GHz, 32 GB 4800 MHz DDR5, 1 TB M.2 SSD, and running 64-bit Window 11 operating system (professional edition).

The true random numbers used in session key generation are obtained from Random.org [48] (which is an open organization to produce the highest quality true random numbers).

All test images with different sizes come from open image database. Part of the encryption and decryption results are demonstrated in the second and third column of Figure 8 respectively. As we can see, the proposed algorithm can successfully deal with images with different sizes.

4.1 Histogram analysis

The uniform distribution is a typical characteristic of an ideal cipher image. Thus, we utilize histogram, as defined in Equation 15, to observe the changes of distributions between original and cipher images, then the results of histogram tests are presented in Figure 9. The comparison between the second and fourth columns of Figure 9 reveals that, regardless of the fluctuation in the histogram of an original image, it becomes uniformly distributed after encryption.

$$\chi^2 = \sum_{i=0}^{L} \frac{(f_i - p_i)^2}{p_i} \tag{15}$$

As an 8-bit gray pixel, it has 255 possible values indicating that $L \in [0, 255]$. Then, the f_i represents the true proportion of the pixel value i in a cipher image, while p_i represents its expected proportion. The ideal cipher image should have an expected χ^2 value of 293.24783 when the significance level α is set to 0.05. The χ^2 test results, as presented in Table 2, demonstrate that all cipher images successfully pass the test, aligning with the findings of the preceding histogram analysis. Consequently, the proposed algorithm effectively withstands diverse attacks rooted in frequency analysis.

TABLE 6 Results of Local Shannon entropy.

Image	Value	Result
Cameraman	7.902837089	Pass
Boatman	7.902414085	Pass
Train	7.902919915	Pass
Flowers	7.901959679	Pass
Kodim04	7.902546647	Pass
Peppers	7.902752219	Pass
Ruler	7.909859427	Pass
Aerial	7.905918222	Pass
Brain	7.903089546	Pass
Finger	7.903551264	Pass
Kodim01	7.902257919	Pass
Airplane	7.902550321	Pass

4.2 Statistical analysis

Sufficient level of randomness is a fundamental prerequisite to effectively counter attacks that rely on statistical analysis. To comprehensively validate the resistance against statistical attacks, we utilize the SP 800-22 randomness test suite, comprising of 15 tests endorsed by NIST [46], on both cipher images and key streams. The P-values of all randomness tests significantly exceed the significant level (let $\alpha = 0.01$), as indicated in Table 3. Therefore, it can be concluded that both the cipher images and key streams exhibit a high degree of statistical randomness. Hence, attacker cannot rely on statistical analysis to obtain useful information facing the proposed algorithm.

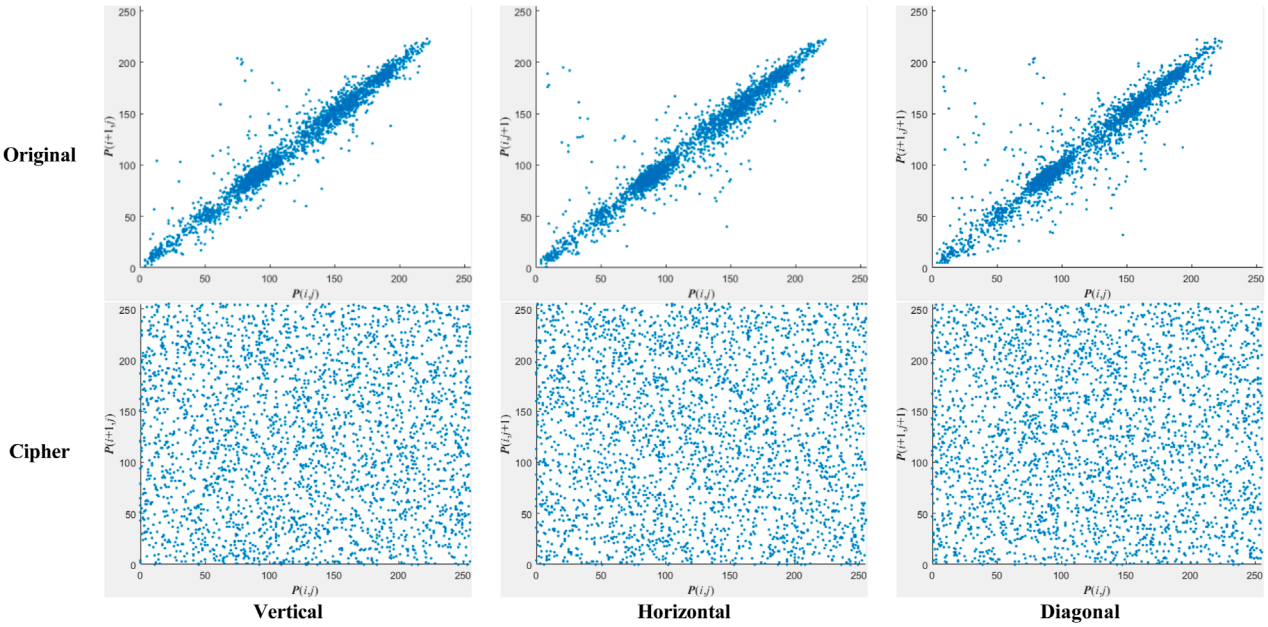

FIGURE 10
Scatter diagram of original image "Peppers" in the first row, and of cipher image "Peppers" in the second row.

4.3 Information entropy analysis

The information entropy, as defined in Equation 16, is commonly employed as a metric for quantifying the level of uncertainty inherent in data. As an 8-bit digital image, the ideal entropy of 8 after encryption will protect it from entropy attacks.

$$H(g) = -\sum_{i=1}^{L} p(g_i) \log_2 p(g_i) \tag{16}$$

The experimental results of the proposed algorithm are presented in Table 4, whereas Table 5 demonstrates the comparative analysis with other algorithms. The proposed algorithm effectively enhances the entropy of any original image to exceeding 7.99, comparable to other established algorithms, which provide sufficient security to resist attacks based on information entropy analysis.

The information entropy is a reliable metric for assessing image uncertainty, but it is important to acknowledge the potential risk of local information leakage. Hence, we further employ a more stringent metric called local information entropy [49], which is computed using Equation 17, to evaluate cipher images.

$$\overline{H}_{k,Tb}(S) = \sum_{i=1}^{k} \frac{H(S_i)}{k} \tag{17}$$

When set image blocks $k = 30$ and non-overlapping pixels $T_B = 1936$, an ideal average local information entropy of S_i should be 7.902469317. However, since all results in Table 6 fall within the confidence interval of 7.901901305–7.903037329, it can be confirmed that the proposed algorithm effectively safeguards images against local entropy analysis based attacks.

4.4 The correlation between adjacent pixels analysis

To provide comprehensive and intricate information, image data contains a significant amount of redundancy, resulting in high correlation between adjacent pixels. Thus, the primary objective of image encryption is to eliminate these correlations to render the image undecipherable. The image 'Peppers' serves as an example. The first row of Figure 10 illustrates that the strong correlations confine most pixels to a specific area in all directions. Subsequently, in the second row of Figure 10, when applying the proposed algorithm, the strong internal relationships within the image are disrupted, resulting in pixels scattering randomly throughout.

The proposed algorithm is further evaluated by an important indicator (γ_{xy}, correlation coefficient) to characterize the effect of correlation reduction. Suppose x_i and y_i are two adjacent pixels, then the calculation of γ_{xy} is given by Equation 18

$$\begin{aligned} \overline{x} &= \frac{1}{N}\sum_{i=1}^{N} x_i, \overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \\ \gamma_{xy} &= \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\left(\sum_{i=1}^{N}(x_i - \overline{x})^2\right)\left(\sum_{i=1}^{N}(y_i - \overline{y})^2\right)}} \end{aligned} \tag{18}$$

By randomly selecting 10,000 pairs of vertically, horizontally, and diagonally adjacent pixels, the γ_{xy} tests are conducted both on original images and their corresponding cipher images. The proposed algorithm effectively minimizes the correlations in the

TABLE 7 Results of γ_{xy} tests.

Direction		Vertical	Horizontal	Diagonal
Cameraman	Original	0.9539	0.9195	0.9015
	Cipher	−0.0057	−0.0044	−0.0019
Boatman	Original	0.9782	0.9779	0.9677
	Cipher	−0.0018	−0.0015	−0.0032
Train	Original	0.9493	0.9583	0.9335
	Cipher	0.0011	−0.0061	1.8×10^{-5}
Flowers	Original	0.9557	0.9567	0.9269
	Cipher	2.9×10^{-4}	−0.0052	8.2×10^{-4}
Kodim04	Original	0.9666	0.9568	0.9431
	Cipher	7.5×10^{-4}	−0.0023	0.0017
Peppers	Original	0.9764	0.9733	0.9651
	Cipher	−0.0028	−0.0019	-4.2×10^{-4}
Ruler	Original	0.4599	0.4494	−0.0291
	Cipher	−0.0020	0.0022	4.3×10^{-4}
Aerial	Original	0.8549	0.8993	0.8003
	Cipher	−0.0013	0.0025	7.6×10^{-4}
Brain	Original	0.9583	0.9467	0.9185
	Cipher	−0.0040	−0.0028	0.0028
Finger	Original	0.9037	0.8869	0.8119
	Cipher	−0.0016	0.0012	3.6×10^{-4}
Kodim01	Original	0.8306	0.8854	0.7609
	Cipher	0.0027	-3.7×10^{-4}	-9.4×10^{-4}
Airplane	Original	0.9458	0.9647	0.9442
	Cipher	0.0016	−0.0017	-4.8×10^{-4}

cipher image, regardless of the high γ_{xy} value of the original images, as demonstrated in Table 7. Thus, attacker cannot perform attacks by analyzing correlation coefficient.

The comparisons with other works are presented in Table 8. It can be observed that the proposed algorithm demonstrates equivalent or even superior ability to conceal correlations between adjacent pixels compared to existing algorithms.

4.5 Plaintext sensitivity analysis

In general, the differential attacks pose the greatest threat to a cryptosystem. To cope with this problem, the proposed algorithm must be extremely sensitive to plaintext changes. It means that the slightest alteration in the original image will result in significant changes to the cipher image. There are two indictors, NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity), to quantitative analyze the plaintext sensitivity of a cryptosystem (Suppose C_1 and C_2 are two cipher images with one bit difference in their corresponding original images), which are given by Equations 19–21

$$D(i,j) = \begin{cases} 0, \boldsymbol{C}_1(i,j) = \boldsymbol{C}_2(i,j) \\ 1, \boldsymbol{C}_1(i,j) \neq \boldsymbol{C}_2(i,j) \end{cases} \tag{19}$$

$$\text{NPCR} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} \boldsymbol{D}(i,j) \times 100\%}{H \times W} \tag{20}$$

$$\text{UACI} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} \frac{|\boldsymbol{C}_1(i,j) - \boldsymbol{C}_2(i,j)|}{255}}{H \times W} \times 100\% \tag{21}$$

The NPCR/UACI tests are conducted 100 times for each original image by randomly selecting a pixel and modifying its least significant bit. As illustrated in Table 9, all results of NPCR and UACI closely match their expected values of 99.6094% and 33.4635%, exhibiting the excellent security performance of the proposed algorithm in countering differential attacks.

Moreover, the proposed algorithm exhibits superior resistance to differential attacks compared to other algorithms as demonstrated in Table 10.

4.6 Key sensitivity analysis

In general, the session key should be the sole confidential element within a practical cryptosystem. Thus, we introduce a true random number to ensure the unpredictability during the generation of session key. Furthermore, the cryptosystem must be sensitive to the session key in two aspects: On one hand, even slight modifications to the session key can have a significant impact on the cipher image; on the other hand, minor errors in the session key cannot yield any useful information from the cipher image.

To verify the key sensitivity performance of the proposed algorithm, we take a used session key *SK* = {035bb58efb252fcc69a95758ea791b59b7f8b271e763e1a14bec4fef54dae437} as a test object. As described in *Step A1-A3*, the round keys will be computed from the session key. Subsequently, they will be further divided into 12 parts to generate initial values for chaotic systems. Thus, we also extract 12 parts from *SK* and apply NPCR/UACI tests. Hint: The original byte is highlighted in green, while the modified one is highlighted in red, with only a single bit being flipped in each test. Firstly, the key sensitivity test results in encryption end are listed in Table 11. Obviously, the proposed algorithm can effectively transform the original image into distinct cipher images using session keys with only one-bit difference, demonstrating its sufficient sensitivity to the session key in encryption end.

TABLE 8 Comparisons with other works.

Image	Direction	Ours	Reference [33]	Reference [34]	Reference [35]	Reference [36]	Reference [38]	Reference [39]
Cameraman	Vertical	−0.0057	−0.0486	—	0.0017	0.0026	—	0.0039
	Horizontal	−0.0044	0.0481	—	−0.0005	-1.9×10^{-4}	—	0.0014
	Diagonal	−0.0019	0.0026	—	0.0021	0.0020	—	−0.0098
Peppers	Vertical	−0.0028	−0.0486	−0.0005	-	-7.1×10^{-4}	−0.0182	−0.0008
	Horizontal	−0.0019	0.0481	0.0009	-	−0.0045	0.0032	0.0038
	Diagonal	-4.2×10^{-4}	0.0026	0.0017	-	−0.0014	−0.0021	0.0076

TABLE 9 Results of NPCR and UACI test (Unit: %).

Image	Cameraman	Boatman	Train	Flowers	Kodim04	Peppers	Ruler	Aerial	Brain	Finger	Kodim01	Airplane
$\overline{\text{NPCR}}$	99.6105	99.5998	99.5950	99.6139	99.6020	99.5827	99.5984	99.6031	99.5999	99.6135	99.6102	99.6147
$\overline{\text{UACI}}$	33.4722	33.4697	33.4627	33.4620	33.4539	33.4546	33.4601	33.4628	33.4594	33.4701	33.4676	33.4586

TABLE 10 Comparisons with other algorithms (Unit: %).

Image	Test	Ours	Reference [34]	Reference [35]	Reference [36]	Reference [37]	Reference [39]
Cameraman	$\overline{\text{NPCR}}$	99.6105	-	99.6067	-	99.6166	99.6142
	$\overline{\text{UACI}}$	33.4722	-	33.4467	-	33.4784	33.4656
Peppers	$\overline{\text{NPCR}}$	99.5827	99.6103	-	99.6017	99.6166	99.6142
	$\overline{\text{UACII}}$	33.4546	33.3215	-	33.4638	33.4784	33.4656

Next, the above modified session keys are served as error session keys to decrypt cipher image. Take image 'Peppers' as an example, the decrypted images are shown in Figure 11, and the results of NPCR/UACI tests are listed in Table 12. Hence, attackers cannot obtain any useful information by trying different session keys even being close to the correct session key.

4.7 Key space analysis

The session key in this paper is derived by combining a 256-bit true random number with the hash value of all original images. Subsequently, round keys can be computed based on the session key to generate initial values for chaotic systems. The above experimental results demonstrate that any alteration in the session key can potentially impact the encryption/decryption outcomes. Consequently, the proposed algorithm ensures a 256-bit key space.

Some literature suggest that a larger key space enhances the overall security of a cryptosystem. However, an excessively large key space also introduces additional challenges in terms of key exchange. In this paper, the 256-bit key space surpasses the theoretically secure 128-bit key space [50, 51], thus validating our algorithm's ability to effectively withstand exhaustive attacks while maintaining efficient session key transmission.

4.8 Chosen-plaintext attack analysis

The attackers may exhaust all methods to compromise a cryptosystem, with chosen-plaintext attacks (CPA) being considered the most potent and practical tool in this regard. To perform CPA on image cryptosystems, a particular image with all pixel values being 0 (naming image 'Black') or 255 (naming image 'White') is consistently considered an optimal carrier. The security performances of the proposed algorithm in handling the image 'Black' and 'White' are comprehensively evaluated based on the above analyses. The results depicted in Figure 12 and Table 13 demonstrate that attackers are unable to differentiate such specific images from ordinary ones using any analytical tools. Thus, the proposed algorithm can resist CPA attacks.

TABLE 11 The key sensitivity test results in encryption end (Unit: %).

Original session key	Modified session key	NPCR			UACI		
		Avg	Max	Min	Avg	Max	Min
035BB58EFB	035BB58EFA	99.6086	99.6234	99.5732	33.4610	33.4709	33.4559
252FCC69A9	252FCC69A8	99.5951	99.6208	99.5705	33.4716	33.4736	33.4551
5758EA791B	5758EA791A	99.6070	99.6223	99.5747	33.4691	33.4748	33.4586
59B7F8B271	59B7F8B270	99.6085	99.6223	99.5737	33.4666	33.4741	33.4599
E763E1A14B	E763E1A14A	99.5812	99.6247	99.5739	33.4753	33.4712	33.4598
EC4FEF54DA	EC4FEF54DB	99.6150	99.6247	99.5721	33.4781	33.4802	33.4563
EF54DAE437	EF54DAE436	99.5969	99.6227	99.5726	33.4688	33.4716	33.4572
E1A14BEC4F	E1A14BEC4E	99.6147	99.6239	99.5748	33.4692	33.4711	33.4570
F8B271E763	F8B271E762	99.6097	99.6237	99.5832	33.4508	33.4711	33.4552
EA791B59B7	EA791B59B6	99.5929	99.6204	99.5751	33.4600	33.4729	33.4584
CC69A95758	CC69A95759	99.5802	99.6215	99.5759	33.4660	33.4704	33.4555
B58EFB252F	B58EFB252E	99.6021	99.6218	99.5726	33.4723	33.4802	33.4557

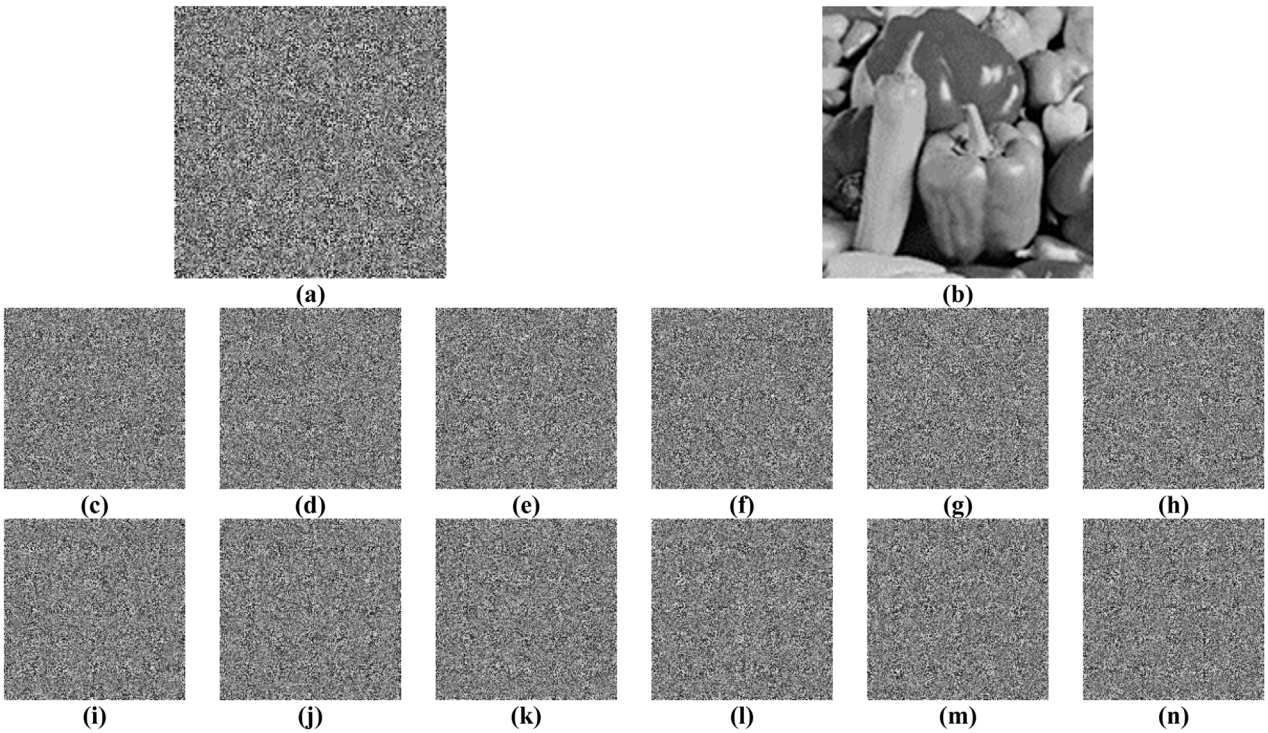

FIGURE 11
The decrypted results of **(A)** cipher image 'Peppers' with **(B)** the correct session key, and with bit flipped at **(C)** *SK*(40), **(D)** *SK*(80), **(E)** *SK*(120), **(F)** *SK*(160), **(G)** *SK*(200), **(H)** *SK*(240), **(I)** *SK*(56), **(J)** *SK*(96), **(K)** *SK*(136), **(L)** *SK*(176), **(M)** *SK*(216), **(N)** *SK*(256).

TABLE 12 The key sensitivity test results in decryption end (Unit: %).

Correct session key	Wrong session key	NPCR			UACI		
		Avg	Max	Min	Avg	Max	Min
035BB58EFB	035BB58EFA	99.6115	99.6183	99.5813	33.4600	33.4691	33.4533
252FCC69A9	252FCC69A8	99.6003	99.6248	99.5847	33.4695	33.4690	33.4534
5758EA791B	5758EA791A	99.6120	99.6188	99.5890	33.4664	33.4679	33.4529
59B7F8B271	59B7F8B270	99.6147	99.6186	99.5853	33.4657	33.4691	33.4541
E763E1A14B	E763E1A14A	99.6072	99.6232	99.5836	33.4614	33.4657	33.4549
EC4FEF54DA	EC4FEF54DB	99.6118	99.6202	99.5831	33.4692	33.4648	33.4546
EF54DAE437	EF54DAE436	99.6100	99.6165	99.5891	33.4569	33.4692	33.4515
E1A14BEC4F	E1A14BEC4E	99.5976	99.6150	99.5800	33.4565	33.4622	33.4546
F8B271E763	F8B271E762	99.6088	99.6196	99.5874	33.4555	33.4622	33.4528
EA791B59B7	EA791B59B6	99.6129	99.6237	99.5865	33.4624	33.4629	33.4519
CC69A95758	CC69A95759	99.6009	99.6178	99.5885	33.4599	33.4622	33.4546
B58EFB252F	B58EFB252E	99.5997	99.6196	99.5875	33.4636	33.4682	33.4520

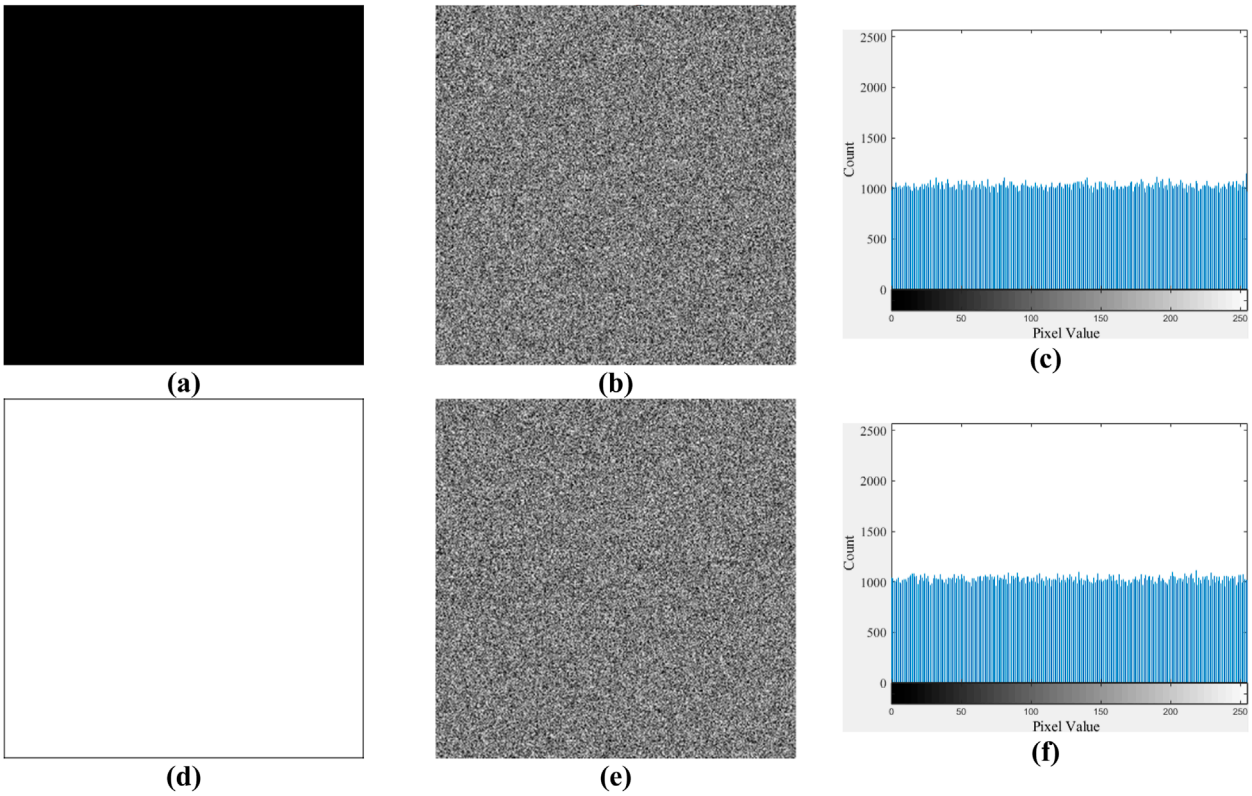

FIGURE 12
Encryption results and histogram analysis of image **(A)** "Black" and **(D)** "White", shown in the first row and second row respectively.

TABLE 13 Experimental results for images 'Black' and 'White'.

Image	χ^2 test	$\overline{\text{NPCR}}$(%)		$\overline{\text{UACI}}$(%)		Entropy		γ_{xy}		
		Plaintext	Key	Plaintext	Key	Global	Local	Horizontal	Vertical	Diagonal
Black	0.9998	99.6109	99.6151	33.4737	33.4762	7.999	7.9026	0.0002	−0.0001	−0.0014
White	0.9961	99.6069	99.6094	33.4713	33.4583	7.999	7.9028	−0.0009	−0.0018	−0.0009

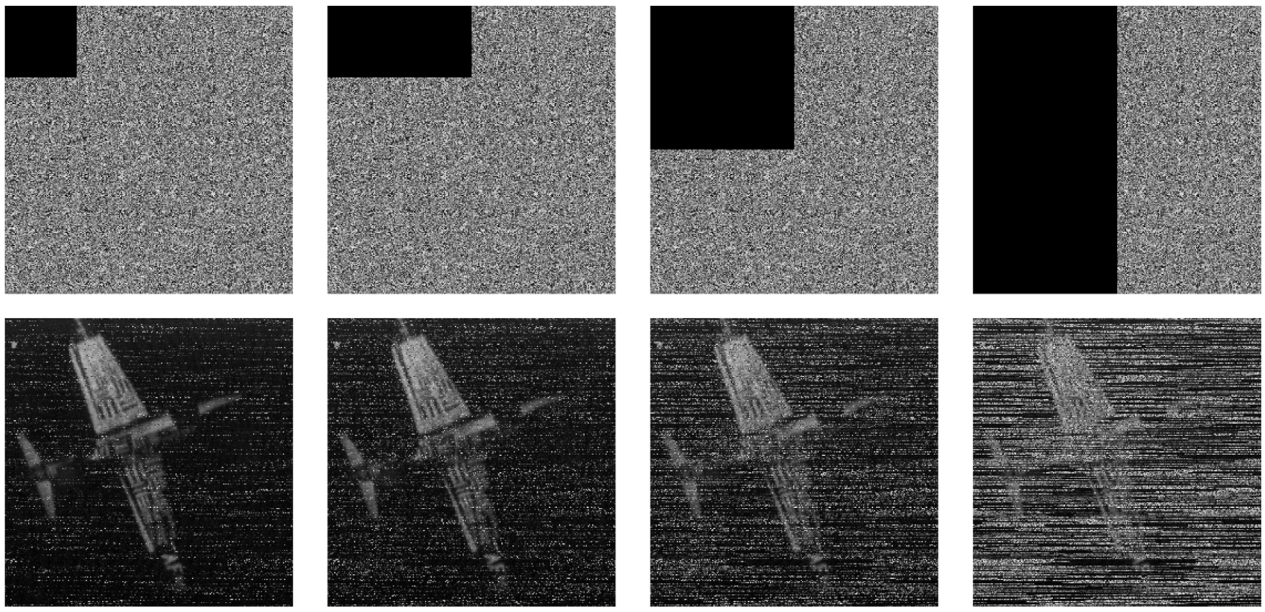

FIGURE 13
Cipher images with data loss 1/16, 1/8, 1/4, 1/2 in the first row and their decrypted image in the second row.

4.9 Robustness analysis

The public communication channel often experiences disturbances like noise and data loss, but images remain useable even with some pollution in daily use. Therefore, the ability of an image cryptosystem to decrypt corrupted cipher images becomes crucial.

Taking the image 'Airplane' as an example, we sequentially cropped 1/16, 1/8, 1/4, and 1/2 portions and subsequently decrypted them using the appropriate session key. As illustrated in Figure 13, although there is some loss of data in the decrypted images, they can still be discerned as representing the ' Airplane ' image to a certain extent. The recognition of the decrypted image becomes increasingly challenging with higher levels of data loss during transmission.

The image 'Airplane' is subjected to Salt&Pepper noise and Gaussian noise, which are two common types of noise. Subsequently, the correct session key is used to decrypt the image. As depicted in Figure 14, the proposed algorithm successfully restores a recognizable image even when it is contaminated by different types of noise with varying degrees. Therefore, we can assert that the proposed algorithm exhibits resilience against minor noise interference in a vulnerable channel.

4.10 Computational cost analysis

The practicality of encryption algorithms is not solely determined by security, it also relies on high operational efficiency, which should be considered a crucial criterion. In this paper, by introducing multi-thread technique, four images (even with different sizes) can be parallel encrypted on four threads using a single image encryption algorithm. Moreover, the counter mode is utilized to derive round keys from a session key that is associated with all original images, to effectively address the issue of secure key transmission.

As illustrated in Figure 4, the counter mode can generate round keys far before encryption process. Thus, the time complexity of the proposed algorithm depends almost entirely on the following parts: 1) The generation of key streams for diffusion, which has a time complexity of $O(WH)$; 2) the generation of key streams for initial vectors, which has a time complexity of $O(W)$; 3) the generation of key streams for initial scrambling, which has a time complexity of $O(W + H)$; and 4) the Algorithm SSDIEA, which has a time complexity of $O(2W+2H)$. Overall, the time complexity of the proposed algorithm is square order $O(WH)$. However, due to the linear order time complexity $O(2W+2H)$

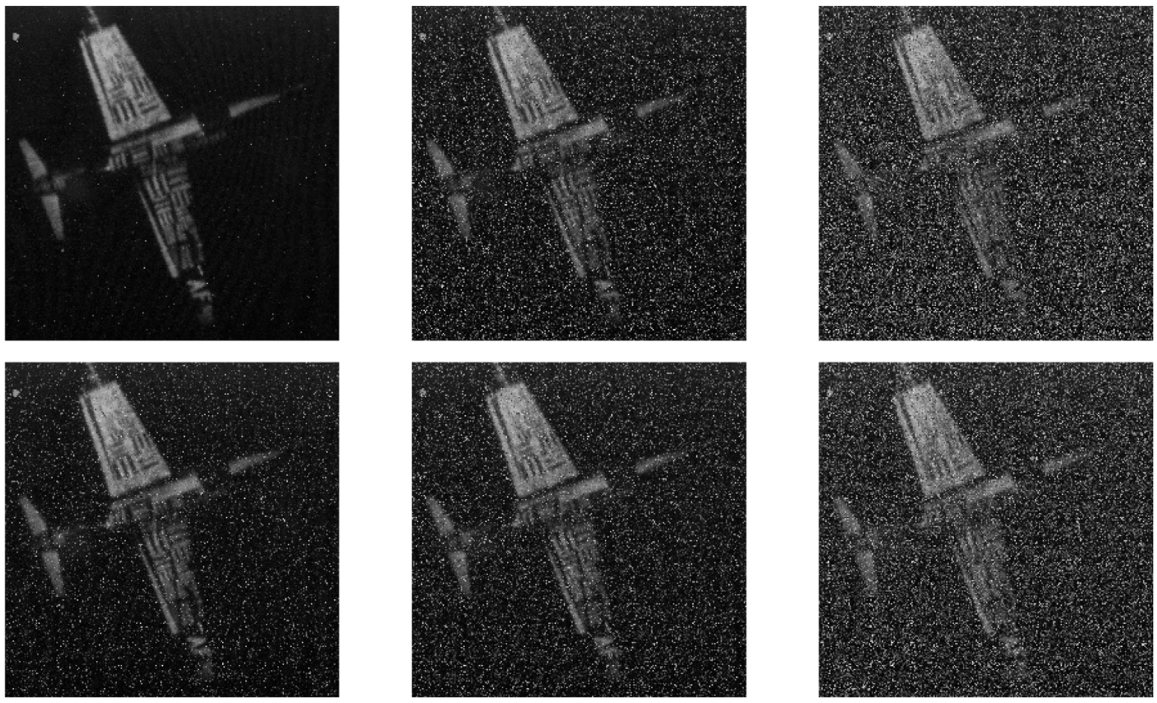

FIGURE 14
Decrypted image from cipher images polluted with Gaussian noise of 0.001, 0.005, 0.01 in the first row, and with Salt&Pepper noise of 0.01, 0.05; (f) 0.1 in the second row.

TABLE 14 Results of time consumption tests (Unit: second).

Images	Ours	Reference [33]	Reference [34]	Reference [35]	Reference [36]	Reference [39]
Four 256 × 256 images	0.0533	—	0.95	0.2088	0.16	0.094
Four 512 × 512 images	0.1495	—	2.71	0.6617	0.41	0.321
100 images with different sizes	2.1046	—	—	—	—	—
Time complexity of encryption process	$O(2W+2H)$	$O(WH)$	$O(WH)$	$O(WH)$	$O(WH)$	$O(WH)$

of the Algorithm SSDIEA, the proposed algorithm presents the superior speed advantage compared with other algorithms, as shown in Table 14.

5 Conclusion

This paper presents a novel batch images encryption algorithm using the counter mode and a multi-channel processing scheme. Firstly, by combining the hash value of all original images with a 256-bit true random number, the proposed algorithm's session key not only exhibits ample sensitivity to plaintext but also possesses strong unpredictability. Subsequently, the counter mode is introduced to generate round keys for encrypting each four images, thereby significantly simplifying session key management when dealing with massive images. Secondly, a multi-channel pseudo random number generator is constructed using two discrete hyperchaotic systems (2D-SCCM and 2D-SIDCM), which ensures rapid provision of four diffusion key streams. Moreover, by designing an adapter, the proposed algorithm can easily employ the multi-thread technique to parallel encrypt four different sized images during each encryption round without additional operations on the original images. Thirdly, the encryption speed can be further accelerated by employing a simultaneous scrambling and diffusion image encryption algorithm on each thread, which leverages its vectorized CBC-based structure to efficiently and securely encrypt every single image. Finally, common security analyses are performed on the proposed algorithm. The results of χ^2 and randomness tests are all over the confidential value 0.01, thus the cipher images can resist attacks based on any statistical analysis. The results of information entropy or correlation coefficient tests are close to the ideal value eight or 0, which indicate that the proposed algorithm can sufficiently hide relations between original and cipher images to prevent attacker obtain any useful information from

cipher images. The results of NPCR and UACI tests being close to 99.6094% and 33.4635% reveal that the proposed algorithm has enough sensitivity to both plaintext and session key changes to resist brute-force, differential, or even chosen-plaintext attacks. In addition, due to the multi-thread technique, the proposed algorithm can parallel encrypted four images at once using a simultaneous scrambling and diffusion image encryption with a linear time complexity of $O(2W+2H)$, thus 100 images can be encrypted in about 2.1 s. In the following research, first, we will try to extend the proposed algorithm to batch color images encryption, and dedicate to implement the proposed algorithm on different kinds of IoT devices, such as the Lora node, Zigbee node, to further optimize its security and efficiency performances.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

BG: Conceptualization, Methodology, Software, Validation, Writing–original draft. GQ: Writing–review and editing. ZS: Investigation, Methodology, Writing–review and editing. JL: Conceptualization, Formal Analysis, Software, Writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported by the National Natural Science Foundation of China (62004108) and in part by the Natural Science Foundation of China (No. 62341118); The program of Entrepreneurship and Innovation Ph.D. in Jiangsu Province (JSSCBS20211175). "Huaishang Talent Plan" for Outstanding Doctoral Program in Huai'an Universities (Z302J23212).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Li H, et al. Optimization of graph clustering inspired by dynamic belief systemsKnowl IEEE T. *Data En* (2024) 36(11):6773–85. doi:10.1109/TKDE.2023.3274547

2. Li H, et al. Overlapping graph clustering in attributed networks via generalized cluster potential game. *ACM Trans Knowl Discov Data* (2023) 18(1):1–26. doi:10.1145/3597436

3. Chen J, Li T, Zhang Y, You T, Lu Y, Tiwari P, et al. Global-and-Local attention-based reinforcement learning for cooperative behaviour control of multiple UAVs. *IEEE T Veh Technol* (2024) 73(3):4194–206. doi:10.1109/TVT.2023.3327571

4. Hosseinzadeh R, Zarebnia M, Parvaz R. Hybrid image encryption algorithm based on 3D chaotic system and choquet fuzzy integral. *Opt Laser Technol* (2019) 120:105698. doi:10.1016/j.optlastec.2019.105698

5. Vaidyanathan S, Kammogne AST, Tlelo-Cuautle E, Talonang CN, Abd-El-Atty B, Abd El-Latif AA, et al. A novel 3-D jerk system, its bifurcation analysis, electronic circuit design and a cryptographic application. *Electronics* (2023) 12(13):2818. doi:10.3390/electronics12132818

6. Han B, Jhaveri RH, Wang H, Qiao D, Du J. Application of robust zero-watermarking scheme based on federated learning for securing the healthcare data. *IEEE J Biomed Health* (2023) 27(2):804–13. doi:10.1109/JBHI.2021.3123936

7. Gong L, Luo H. Dual color images watermarking scheme with geometric correction based on quaternion FrOOFMMs and LS-SVR. *Opt Laser Technol* (2023) 167:109665. doi:10.1016/j.optlastec.2023.109665

8. Tan J, Liao X, Liu J, Cao Y, Jiang H. Channel attention image steganography with generative adversarial networks. *IEEE T Netw Sci Eng* (2022) 9(2):2021888–903. doi:10.1109/TNSE.2021.3139671

9. Zhou Z, Wan Y, Cui Q, Yu K, Mumtaz S, Yang CN, et al. Blockchain-based secure and efficient secret image sharing with outsourcing computation in wireless networks. *IEEE T Wirel Commun* (2024) 23(1):423–35. doi:10.1109/TWC.2023.3278108

10. Hsieh PA, Wu J. A review of the asymmetric numeral system and its applications to digital images. *Entropy* (2022) 24(3):e24030375. doi:10.3390/e24030375

11. Kolivand H, Hamood SF, Asadianfam S, Rahim MS. RETRACTED ARTICLE: image encryption techniques: a comprehensive review. *Multimed Tools Appl* (2024) 83:73789. to be published. doi:10.1007/s11042-023-17896-0

12. Guvenoglu E, Tunali V. ZigZag transform with Durstenfeld shuffle for fast and secure image encryption. *Connect Sci* (2023) 35(1):2162000. doi:10.1080/09540091.2022.2162000

13. Lu H, Teng L, Du L. Image encryption with 1D-MS chaotic systems and improved zigzag disambiguation. *Eur Phys J Plus* (2024) 139:350. doi:10.1140/epjp/s13360-024-05146-7

14. Alexan W, Korayem Y, Gabr M, El-Aasser M, Maher EA, El-Damak D, et al. AntEater: when arnold's cat meets langton's ant to encrypt images. *IEEE Access* (2023) 11:106249–76. doi:10.1109/ACCESS.2023.3319335

15. Inam S, Kanwal S, Firdous R, Hajjej F. Blockchain based medical image encryption using Arnold's cat map in a cloud environment. *Sci Rep* (2024) 14(1):5678. doi:10.1038/s41598-024-56364-z

16. Rani N, Sharma SR, Mishra V. Grayscale and colored image encryption model using a novel fused magic cube. *Nonlinear Dyn* (2022) 108(2):1773–96. doi:10.1007/s11071-022-07276-y

17. Ko HJ, Huang CT, Tseng HW, Wang SJ. Efficient cost-reduced with high-quality image of imperceptible steganography using modulo and magic cube. *IEEE Access* (2022) 10:67686–93. doi:10.1109/ACCESS.2022.3185120

18. Bezerra J. I. M., Machado G, Molter A, Soares RI, Camargo V. A novel simultaneous permutation-diffusion image encryption scheme based on a discrete space map. *Chaos Soliton. Fract.* (2023) 168:113160. doi:10.1016/j.chaos.2023.113160

19. Li T., Fan W, Wu J, Zhang D. Image encryption based on a fractional-order hyperchaotic system and fast row-column-level joint permutation and diffusion. *Nonlinear Dyn* (2024) 112:10555–81. doi:10.1007/s11071-024-09597-6

20. Murillo-Escobar M.A., Cruz-Hernández C, Cardoza-Avendaño L, Méndez-Ramírez R. A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. *Nonlinear Dyn.* (2017) 87:407–25. doi:10.1007/s11071-016-3051-3

21. Barani M.J., et al. A new Pseudo random number generator based on generalized Newton complex map with dynamic key. *J. Inf. Secur. Appl.* (2020) 53:102509. doi:10.1016/j.jisa.2020.102509

22. Benkouider K., Sambas A, Bonny T, Al Nassan W, Moghrabi IAR, Sulaiman IM, et al. A comprehensive study of the novel 4D hyperchaotic system with self-exited multistability and application in the voice encryption. *Sci. Rep.* (2024) 14:12993. doi:10.1038/s41598-024-63779-1

23. Hua Z., Zhu Z, Chen Y, Li Y. Color image encryption using orthogonal Latin squares and a new 2D chaotic system. *Nonlinear Dyn.* (2021) 104:4505–22. doi:10.1007/s11071-021-06472-6

24. Gao X., Sun B, Cao Y, Banerjee S, Mou J. A color image encryption algorithm based on hyperchaotic map and DNA mutation. *Chin. Phy. B* (2023) 32(3):030501. doi:10.1088/1674-1056/ac8cdf

25. Huang X., Dong Y, Ye G, Shi Y. Meaningful image encryption algorithm based on compressive sensing and integer wavelet transform. *Front. Comput. Sci.* (2023) 17(3):173804. doi:10.1007/s11704-022-1419-8

26. Teng L., Wang X., Xian Y. Image encryption algorithm based on a 2D-CLSS hyperchaotic map using simultaneous permutation and diffusion. *Inform. Sci.* (2022) 605:71–85. doi:10.1016/j.ins.2022.05.032

27. Huang Z., Zhou N. Image encryption scheme based on discrete cosine Stockwell transform and DNA-level modulus diffusion. *Opt. Laser. Technol.* (2022) 149:107879. doi:10.1016/j.optlastec.2022.107879

28. Sambas A., Miroslav M, Vaidyanathan S, Ovilla-Martínez B, Tlelo-Cuautle E, El-Latif AAA, et al. A New Hyperjerk System With a Half Line Equilibrium: Multistability, Period Doubling Reversals, Antimonotonocity, Electronic Circuit, FPGA Design, and an Application to Image Encryption. *IEEE Access* (2024) 12:9177–94. doi:10.1109/ACCESS.2024.3351693

29. Li Y., Wang Q., Yu S. A novel hybrid scheme for chaotic image encryption. *Phys. Scr.* (2024) 99(4):045244. doi:10.1088/1402-4896/ad3171

30. Sangavi V., Thangavel P. An exquisite multiple image encryption harnessing multi-scroll Lu–Chen and Chua chaotic systems employing domino strategy. *J. Inf. Secur. Appl.* (2023) 72:103408. doi:10.1016/j.jisa.2022.103408

31. Kumar Y., Guleria V. A novel multiple image encryption technique based on asymmetric cryptosystem with HCM in frequency domain. *Multimed. Tools Appl.* (2024) 83:72253–78. to be published. doi:10.1007/s11042-024-18347-0

32. Morteza S. K., Amirabbas G., Mehdi Y. A comprehensive survey on image encryption: Taxonomy, challenges, and future directions. *Chaos Soliton. Fract.* (2024) 178:114361. doi:10.1016/j.chaos.2023.114361

33. Sahasrabuddhe A., Laiphrakpam D. S. Multiple images encryption based on 3D scrambling and hyper-chaotic system. *Inform. Sciences* (2021) 550:252–67. doi:10.1016/j.ins.2020.10.031

34. Zhang X., Liu M., Tian J. Multiple-image encryption algorithm based on Sarrus rule and 3D Fibonacci matrix. *Phys. Scr.* (2023) 98:055208. doi:10.1088/1402-4896/acc905

35. Zhou Z., Xu X, Yao Y, Jiang Z, Sun K Novel multiple-image encryption algorithm based on a two-dimensional hyperchaotic modular model. *Chaos Soliton. Fract.* (2023) 173:113630. doi:10.1016/j.chaos.2023.113630

36. Wang X., Liu H. Cross-plane multi-image encryption using chaos and blurred pixels. *Chaos Soliton. Fract.* (2022) 164:112586. doi:10.1016/j.chaos.2022.112586

37. Tao L., Liang X, Hu B, Han L Compound encryption of multiple images by utilizing a novel chaos and nonlinear transform. *Neural Comput.&lic., DPTA-* (2022) 2021. doi:10.1007/s00521-022-07849-3

38. Xiao D., Zhao M., Wang M. Low-cost and secure multi-image encryption scheme based on P-tensor product compressive sensing. *Opt. Laser. Technol.* (2021) 140:107077. doi:10.1016/j.optlastec.2021.107077

39. 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) 108:613–36. doi:10.1007/s11071-021-07192-7

40. Liu X., Wang J., Li J. URTSegNet: A real-time segmentation network of unstructured road at night based on thermal infrared images for autonomous robot system. *Control Eng. Pract.* (2023) 137:105560. doi:10.1016/j.conengprac.2023.105560

41. Chen J., Du C, Zhang Y, Han P, Wei W A Clustering-Based Coverage Path Planning Method for Autonomous Heterogeneous UAVs. *IEEE T. Veh. Technol.* (2022) 23(12):25546–56. doi:10.1109/TITS.2021.3066240

42. Zhou S., Qiu Y, Wang X, Zhang Y Novel image cryptosystem based on new 2D hyperchaotic map and dynamical chaotic S-box. *Nonlinear Dyn* (2023) 111:9571–89. doi:10.1007/s11071-023-08312-1

43. Wen J., Xu X, Sun K, Jiang Z, Wang X Triple-image bit-level encryption algorithm based on double cross 2D hyperchaotic map. *Nonlinear Dyn.* (2023) 11:6813–38. doi:10.1007/s11071-022-08158-z

44. Yuan F., Li Y., Wang G. A universal method of chaos cascade and its applications. *Chaos* (2021) 31(2):021102. doi:10.1063/5.0041518

45. El-Semary A.M., Abdel-Azim M. M. Counter Chain: A New Block Cipher Mode of Operation. *J. Inf. Process. Syst.* (2015) 11:266–279. doi:10.3745/JIPS.03.0031

46. Haider T., Blanco S.A., Hayat U. A novel pseudo-random number generator based on multivariable optimization for image-cryptographic applications. *Expert. Syst. Appl.* (2024) 240:122446. doi:10.1016/j.eswa.2023.122446

47. Haahr M. RANDOM.ORG: True Random Number Service. Available from: https://www.random.org (Accessed August 5, 2024).

48. Pareschi F., Rovatti R., Setti G. On Statistical Tests for Randomness Included in the NIST SP800-22 Test Suite and Based on the Binomial Distribution. *IEEE T. Inf. Foren. Sec.* (2012) 7(2):491–505. doi:10.1109/TIFS.2012.2185227

49. Wu Y., Zhou Y., Saveriades G., Agaian S, Noonan JP, Natarajan P Local Shannon entropy measure with statistical tests for image randomness. *Inform. Sciences* (2013) 222:323–42. doi:10.1016/j.ins.2012.07.049

50. Liu L., Lei Y., Wang D. A fast chaotic image encryption scheme with simultaneous permutation-diffusion operation. *IEEE Access* (2020) 8:27361–74. doi:10.1109/ACCESS.2020.2971759

51. Murillo-Escobar M.A., Meranza-Castillón MO, López-Gutiérrez RM, Cruz-Hernández C Suggested Integral Analysis for Chaos-Based Image Cryptosystems. *Entropy* (2019) 21(8):815. doi:10.3390/e21080815