



OPEN

A color image encryption scheme utilizing a logistic-sine chaotic map and cellular automata

Shiji Sun^{1,2}, Wenzhong Yang^{1,2}, Yabo Yin^{1,2}, Xiaodan Tian^{1,2}, Guanghan Li^{1,2} & Xiangxin Deng^{1,2}

The rapid advancement of 5G technology has led to a significant increase in the generation and transmission of visual image data, underscoring the growing need for robust privacy protection. However, existing chaotic encryption methods often suffer from simplicity and limited security because they rely on single-layer encryption approaches. To address these challenges, this paper introduces a multi-layer image encryption algorithm that leverages the Logistic-Sine chaotic Map (LSCM) and cellular automata for enhanced security. Initially, the R, G, and B channels of an image are extracted and subjected to independent row and column transformations, creating a new, scrambled matrix. Subsequently, the proposed LSCM, which integrates logistic and sine maps to overcome periodic vulnerabilities, generates a rich chaotic sequence for XOR-based encryption. Finally, cellular automata further enhance the obfuscation, increasing the algorithm's complexity and attack resistance. Experimental results demonstrate that the image encryption algorithm achieves high security, strong robustness against noise and data loss, and superior performance in statistical and correlation analyses. These findings suggest that the algorithm is well-suited for protecting visual data in networked environments, offering significant application value in image privacy and security.

Keywords Chaotic encryption, Logistic-sine map, Image encryption, Cellular automata, Multi-layer encryption, Privacy protection

As a result of the growing popularity of the Internet and social network platforms, large amounts of image and video data are now transported through a wide range of applications, including webinars, videoconferences, conferences, online courses and social media platforms. Information security and privacy protection is a primary concern to avoid unauthorized access to data shared on these platforms. There are many techniques available to protect data from intruders such as watermarking¹, Steganography² and cryptography^{3–5}. Encryption is considered as the best method to protect visual data. Encryption transforms visual data into an unrecognizable format, protecting it from eavesdropping. In video and image data, there is inherently a significant correlation and redundancy between pixels. This renders traditional encryption methods such as AES image encryption, TDES image encryption, etc., unsuitable for safeguarding such content. Classical techniques not only take more time but also have slower processing speed. Therefore, researchers are increasingly turning to chaos theory as an alternative method for video encryption. Chaotic maps can produce different effects with small changes in initial conditions. Chaotic map has become a focal point in encryption strategies. Chaos based encryption methods mainly involve alignment and diffusion techniques. Alignment implies dislocation of data, while diffusion refers to the substitution of data^{6–12}. Compared to one-dimensional chaotic maps, high dimensional chaotic maps typically have more complex structures and better chaotic properties. This makes their chaotic trajectories more difficult to predict, but correspondingly increase their overhead cost.

Researchers have designed numerous chaotic maps^{13–15}. Kumar and Dua¹⁶ used two different chaotic maps to encrypt audio using DNA encoding and dynamic diffusion. Wei et al.¹⁷ proposed a secure image transmission scheme using two chaotic maps, tailored for multimedia sensor environments. The scheme includes SIT-SS, which employs a Tent-Logistic Map for efficient encryption with low hardware requirements, and SIT-SR, which uses dynamic permutation and diffusion based on discrete logarithms to enhance security. Simulation results show improved image reconstruction quality and superior statistical properties compared to existing methods. Erkan et al.¹⁸ used Schaffer function to introduce 2D hyper chaotic system. This mapping was used to design image encryption scheme. Mansoor and Parah¹⁹ introduced hybrid adaptive image encryption scheme using logistic

¹School of Computer Science and Technology (School of Cyberspace Security), Xinjiang University, Urumqi 830046, China. ²Xinjiang Key Laboratory of Multilingual Information Technology, Xinjiang University, Urumqi 830046, China. email: yangwenzhong@xju.edu.cn; yinyabo@xju.edu.cn

map-ping and Tent mapping. The authors also integrated DNA computation into the scheme. Experiments verified the security provided by the method. Ma et al.²⁰ proposed an innovative algorithm based on Tabu Search and Hyperchaos. Their method integrates the Tabu search optimization strategy to improve key generation and hyperchaotic systems to enhance the unpredictability of encryption processes. The approach demonstrates strong performance in resisting statistical and brute-force attacks, making it a valuable contribution to the field of image security. Khokhar et al.²¹ controlled the load frequency of the grid based on one-dimensional CS. When refining the approach, they optimized the mapping method with cosine function. The experimental results show that the method has better convergence, effective control of the running time, and higher sensitivity. Wei et al.²² proposed a robust image encryption algorithm, IEASP, based on a novel 2D-SQPM hyperchaotic map and a pixel fusion strategy. The 2D-SQPM map demonstrated strong chaotic properties. IEASP introduced efficient measures, including a common keystream, pixel fusion, chaotic pixel superposition, and intra-vector quick scrambling, achieving high security and efficiency. Wei et al.²³ also propose a novel multi-channel image encryption algorithm, named MIEA-PRHM, which leverages pixel reorganization and hyperchaotic maps. The MIEA-PRHM algorithm employs two hyperchaotic maps to jointly generate chaotic sequences, ensuring a larger key space and enhanced randomness.

Alexan et al.²⁴ employed the Tent mapping technique. They used it to create chaotic sequences. These chaotic sequences formed the basis of a novel optimality seeking strategy. This strategy aimed to overcome the limitations of traditional optimization algorithms, which often get stuck in local optimal solution. By utilizing chaotic sequences, the proposed approach enabled faster convergence to the global optimum. Given the poor security of one-dimensional chaotic systems, researchers in the field of chaotic cryptography have tried to improve the shortcomings of one-dimensional systems by increasing the dimensionality of chaotic systems. However, increasing multiple control parameters also greatly increases its computational complexity. Numerous researchers^{22,25–28} have dedicated themselves to enhancing the security and feasibility of chaotic image encryption. Moreover, their proposed methods primarily concentrate on two key areas: the development or optimization of chaotic systems and the creation or refinement of encryption procedures.

Therefore, we propose a one-dimensional mapping, and improves the security of images through multilevel encryption. The above studies have proposed various chaotic maps for encrypting images or audio. However, a common issue with these methods is that the encryption layer is often too simplistic, typically used for only a single layer of encryption. This may result in the encrypted image being relatively easy to leak.

Therefore, this paper proposes an image encryption algorithm, the LSCM encryption and multi-layer encryption algorithm based on the traditional chaotic image encryption. The combination of matrix determinant transformation, cellular automata and chaotic map produces an image encryption algorithm that is used to solve the problem of image encryption at multiple levels. The main contributions of this thesis are as follows:

- 1. Before subjecting the image to chaotic encryption, we introduce the determinant transform to preprocess the image. It is beneficial to increase the security of the image and improve the attack resistance.
- 2. We propose the LSCM encryption. This paper combines the logistic map and sine map to address their potential periodic phenomenon under certain parameters. This makes the encryption system less prone to be vulnerable to periodic attacks. It can be well avoided by the combination of two chaotic maps. A single chaotic system may have the risk of limited key space or exposure to specific attacks. And mixing two different chaotic systems can greatly enhance the security of encryption. An attacker needs to breach both systems to successfully decrypt the message. It greatly increases the difficulty of the attack. Fusing two chaotic systems at the same time can generate a richer key space.
- 3. Cellular automata is applied after chaotic encryption to further confuse the sequence. Cellular automata has complex dynamic behavior and spatial structure, and chaotic system also has complex nonlinear properties. Combining them can increase the complexity of the encryption system, making it more difficult for crackers to understand how the encryption algorithm works.

The rest of the paper is organized as follows. Section “[Preliminaries](#)” describes the image encryption algorithm based on the Logistic-Sine Map. Section “[Proposed image encryption algorithm](#)” describes the implementation steps as well as the flow of the algorithm. Section “[Experimental results and analysis](#)” experimental tests and security analysis. Section [Conclusions](#) concludes the article.

Preliminaries

This section describes the methods used in this paper, starting with the determinant transform, followed by the proposed image encryption algorithm, and finally the role of the cellular automata.

Chaotic map

Two classical chaotic maps are first reviewed and then the LSCM proposed in this paper is introduced. Logistic mapping is a common nonlinear dynamical system model, the mathematical expression of one-dimensional logistic mapping in Eq. (1):

$$x_{n+1} = r \times x_n \times (1 - x_n) \quad (1)$$

Where: r is the control parameter, $r \in (0, 4]$ the sequence is in chaotic state only when $3.569945627 < r \leq 4$, the sequence x_n is in chaotic state. Arif²⁹ used logistic map for encryption of images. However, the application of this algorithm may only be suitable for encrypting grayscale images, as the algorithm in question is specifically designed for encrypting grayscale images. It may not be able to encrypt well for color images. There is still room

for improvement. Sine chaotic map is a nonlinear dynamical system model based on sinusoidal functions, and the expression of Sine chaotic map is Eq. (2):

$$x_{i+1} = \frac{a}{4} \times \sin(\pi \times x_i) \quad (2)$$

Where a is a control parameter which typically is 4. The range of chaotic orbital state values is (0, 1). Song et al.²⁴ encrypted the image using sine mapping to achieve privacy protection. However, their image encryption algorithm is relatively simple, which may pose a risk of deciphering.

Considering the colour of the encrypted image and the complexity of the encryption process, this paper improves the complexity of the encryption algorithm based on two kinds of mapping: logical mapping and sinusoidal mapping. At the same time, the characteristics of these two mappings are combined to form the image encryption algorithm in this paper. We call it modified Logistic-Sine Chaotic Map (LSCM). Now, we give its expression Eq. (3).

$$x_{i+1} = \sin(r \times \pi \cdot (1 - x_i)) \times b \times x_i \quad (3)$$

Where r and b are the control parameter. when $r \in (0, 4]$. The range of chaotic orbital state value is (0, 1) As in Fig. 1, when r is 4, its distribution value is more random. This chaotic map combines the advantages of both logistic and sine maps. By combining the logistic map with the sine map, it compensates for the periodicity phenomenon that occurs in each map individually under certain parameters. Even slight changes in initial conditions can lead to periodic LSCM trajectories. This chaos models some nonlinear behaviors very well, making its trajectory unpredictable. Additionally, I utilize the Lyapunov exponent (LE) to evaluate its chaotic properties.

The Lyapunov exponent (LE) assesses the chaotic behavior of a dynamical system. A positive LE value indicates that two extremely close LSCM trajectories in the phase plane exponentially separate over time. The system's predictability is determined by the maximum LE (MLE) value. If the MLE value is positive, the system exhibits chaotic behavior. Multiple positive LE values indicate hyperchaotic behavior. Chaotic maps with hyperchaotic behavior are characterized by high complexity and unpredictable LSCM trajectories. Figure 2 shows that the Lyapunov exponent (LE) is greater than 0, indicating good chaotic behavior. The horizontal axis represents the control parameter r , while the vertical axis represents the LE.

Bifurcation maps illustrate variations in chaotic behavior for different parameter values. A larger range of parameters producing chaotic behavior increases resistance to exhaustive attacks. Figure 3 shows the bifurcation diagram of the proposed LSCM chaotic map. Figure 3 shows that the parameter r control produces a high chaotic range and does not produce a periodic window. This indicates good chaotic properties.

Cellular automata

Cellular automata (CA), originally proposed by John von Neumann and Stanislaw Ulam in the 1940s, are a type of discrete model. Cellular automata find applications in various fields, including biology, physics, computer science, and social science. In one-dimensional CA, each cell has only two states, i.e., 0 or 1. For three neighboring

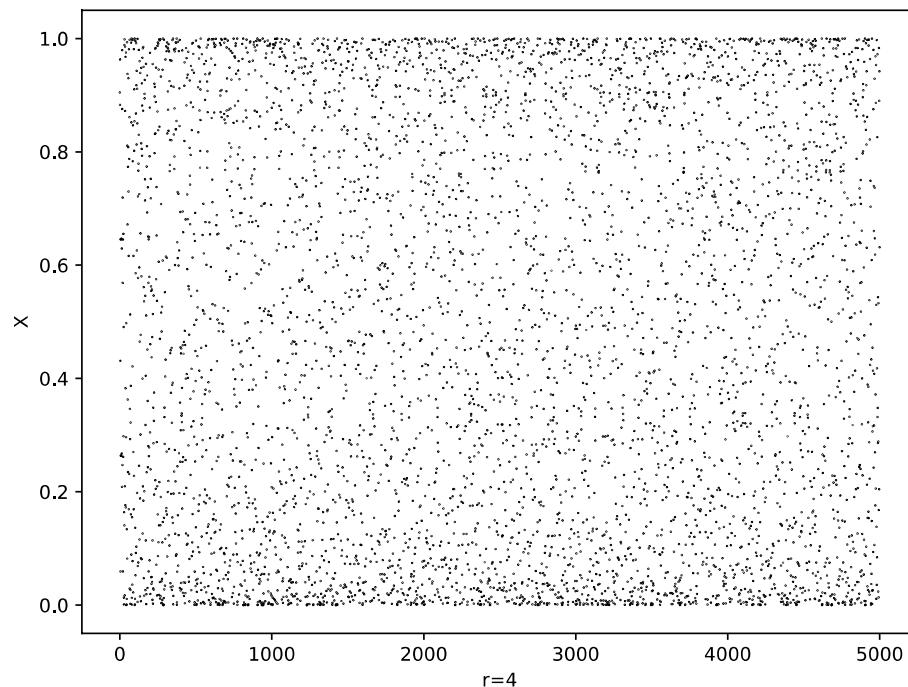


Fig. 1. LSCM trajectories.

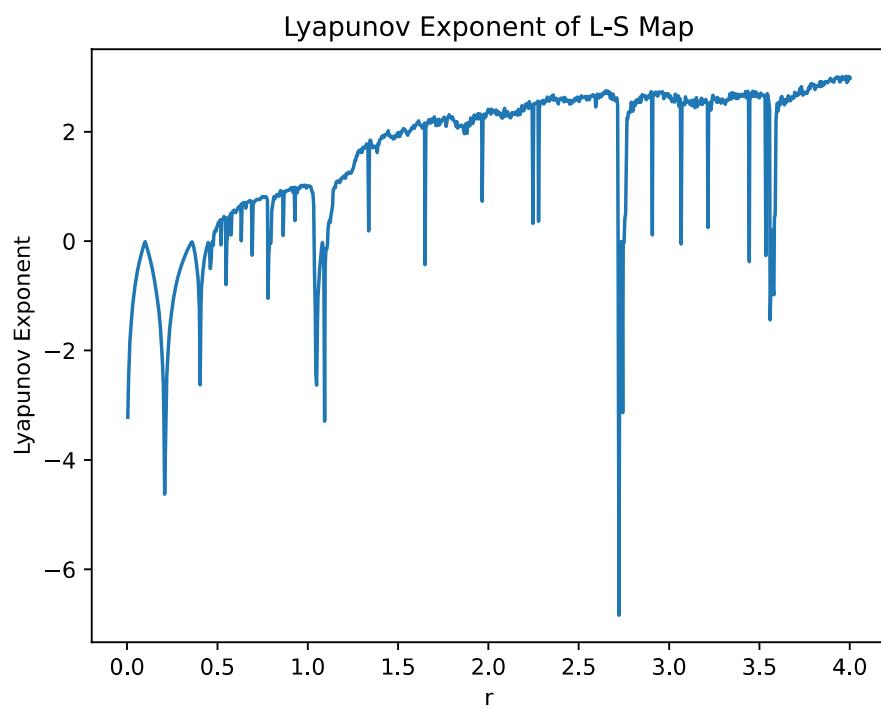


Fig. 2. LE index.

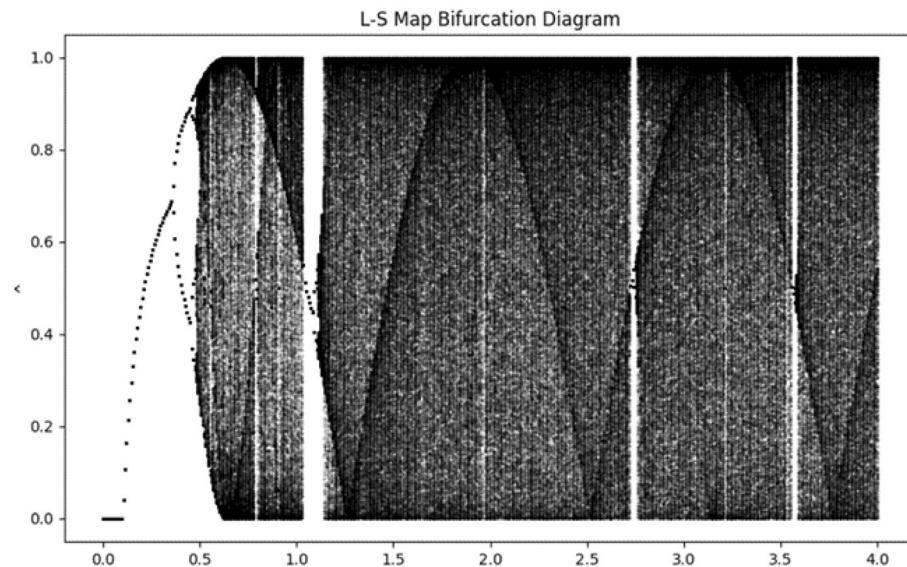


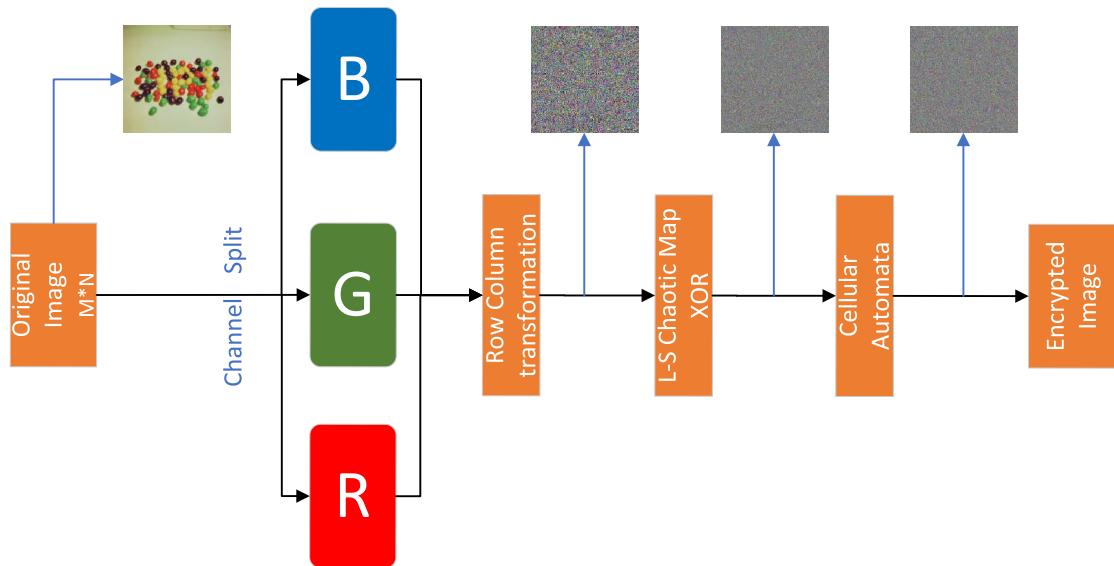
Fig. 3. LS Bifurcation diagram.

cells, there are $2 \times 2 \times 2 = 8$ possible states, which are 000, 001, 010, 011, 100, 101, 110, 111. The state of each cell at the next moment is determined by that cell and its two neighboring cells, as expressed in Eq. (4).

$$S_i^{t+1} = f(S_{i-1}^t, S_i^t, S_{i+1}^t) \quad (4)$$

where: t is time; S_i is the state of the i th cell; S_i^{t+1} is the state of the i th cell at the moment $t + 1$; f is a Boolean function, a binary logic operation. See literature³⁰ for more details. This paper focuses on encrypting images using cellular automata to generate an unordered sequence.

| Test item | LSCM |
|----------------------------------|--------------|
| Frequency (Bit) | 19/20 (95%) |
| Frequency (Block) | 19/20 (95%) |
| Run | 20/20 (100%) |
| Longest Run | 20/20 (100%) |
| Binary Matrix Rank | 20/20 (100%) |
| Discrete Fourier Transform | 20/20 (100%) |
| Non-Overlapping Temp. | 19/20 (95%) |
| Overlapping Temp. | 20/20 (100%) |
| Universal Statistical | 20/20 (100%) |
| Linear Complexity | 19/20 (95%) |
| Serial 1 | 20/20 (100%) |
| Serial 2 | 19/20 (95%) |
| Appr. Entropy | 20/20 (100%) |
| Cum. Sums Forward | 20/20 (100%) |
| Cum. Sums Reverse | 20/20 (100%) |
| Random Excurs. ($x = -1$) | 20/20 (100%) |
| Random Excurs. ($x = +1$) | 20/20 (100%) |
| Random Excurs. Var. ($x = -1$) | 20/20 (100%) |
| Random Excurs. Var. ($x = +1$) | 20/20 (100%) |

Table 1. NIST test results for LSCM.**Fig. 4.** Block diagram of proposed image encryption.

NIST SP-800 randomness test

The NIST SP-800 randomness test is a widely adopted collection of statistical methodologies aimed at assessing the quality of random and pseudorandom number generations. To demonstrate the superiority of our chaotic sequence generation strategy, we subject the chaotic sequences employed within our LSCM to the NIST SP-800 randomness test.²³ Following the NIST SP-800 randomness test requirements, we generate 60 sets of binary sequences, each 1,000,000 bits long. Following the NIST SP-800 randomness test requirements, we generate 60 sets of binary sequences, each 1,000,000 bits long. Table 1 presents the final test results.

Proposed image encryption algorithm

The encrypted image size is denoted as $M * N$. The encryption process, shown in Fig. 4 and implemented by Algorithm 1. The method combines matrix rank transformation, chaotic maps, and cellular automata for multi-level encryption and decryption. The original image has obtained in a size of $M * N$. The image is then converted into a matrix. The original image is divided into R, G, and B channels. The R, G, and B channels are

transformed with different ranks to increase deciphering difficulty. After row and column transformations, the three channels remain independent. A new matrix I is generated. The transformed image undergoes LSCM chaotic encryption. The chaotic sequence, producing values in $(0,1)$, is scaled and matched with the image values. The chaotic sequence is converted to form a $M * N$ matrix, and then XOR operation is performed with the encrypted image matrix to create a new matrix. After chaotic encryption, the image undergoes the cellular automata for final encryption to obtain the final ciphertext image E .

Require: Original Image I
Ensure: Encrypted Image E

- 1: **Step 1: Shuffling (Pixel Position Shuffling)**
- 2: Split I into channels: B, G, R
- 3: **for** each channel (B, G, R) **do**
- 4: Reverse every n -th column (specific pattern)
- 5: Interchange columns between channels (specific pattern)
- 6: Reverse rows and elements
- 7: Apply circular shift to rows
- 8: **end for**
- 9: Merge channels: $I_{\text{shuffled}} \leftarrow \text{Merge}(B, G, R)$
- 10: **Step 2: Chaotic Map Encryption**
- 11: Generate chaotic matrix $K[i, j]$ using LSCM
- 12: **for** each pixel (i, j) in I_{shuffled} **do**
- 13: $I_{\text{chaotic}}[i, j] \leftarrow I_{\text{shuffled}}[i, j] \oplus K[i, j]$
- 14: **end for**
- 15: **Step 3: Cellular Automaton Encryption**
- 16: **for** each channel (B, G, R) in I_{chaotic} **do**
- 17: **for** each pixel (i, j) in channel **do**
- 18: Convert pixel value to binary
- 19: Apply XOR and NOT operations (4 iterations)
- 20: Convert back to decimal
- 21: **end for**
- 22: **end for**
- 23: Merge channels: $E \leftarrow \text{Merge}(B, G, R)$
- 24: **return** E

Algorithm 1. Encryption

Channel split

An image is first divided into three channels, which are then processed separately to enhance security. Figure 5 shows the image after channel separation. The original image has a size of $M * N$. The image is converted into a matrix. The original image is divided into R, G and B channels.

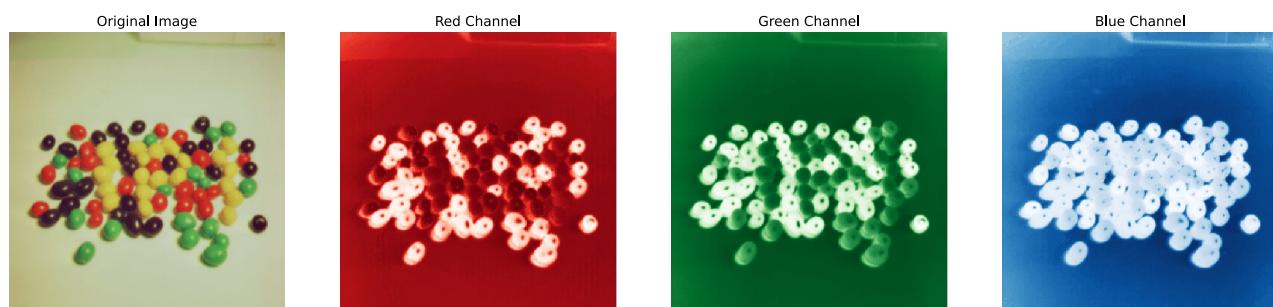


Fig. 5. RGB channel image.

Row and column transformation

In this paper, the image is segmented and encrypted. The encrypted image is split into multiple channels, each undergoing different row and column transformations to generate a new image. The process is as follows.

- Step 1: For an encrypted image, we split it into a number of channels to perform different row and column transformations on it respectively. The transformation starts with the second column. The odd columns of the G channel and the reverse order of the R channel are inverted. Every third column in the B channel is inverted.
- Step 2: The R and G channels are switched in every column. The R and B channels are switched in every third column. G and B channels are switched in every fifth column.
- Step 3: An inversion operation is performed for each channel.
- Step 4: A cyclic shift operation is performed on each row of the R channel, every three rows of the B channel, and every five rows of the G channel.
- Step 5: The row-column transformation is completed.

Generating chaotic sequences

A matrix of the same size as the encrypted image is generated using the Logistic-Sine Chaotic Map (LSCM). The matrix values are scaled to match the pixel value range [0, 256]. Substitution and diffusion operations generate a new encrypted matrix. The LSCM process is as follows:

- **Step 1:** A chaotic matrix I_{chaotic} is generated using LSCM.
- **Step 2:** Scale the values of matrix K to the range [0, 256] to obtain a new matrix K .
- **Step 3:** Diffusion and substitution operations using matrices I_{shuffled} and K generate the encrypted image I_{chaotic} .

Cellular automata process

After chaotic encryption, we further encrypt the cryptographic image matrix using CA to enhance security. And we achieve pixel-level encryption and decryption of the image. The encryption and decryption processes of CA are shown in Fig. 6. Algorithm 2 illustrates the encryption process. The decryption process is the inverse of the encryption algorithm. Firstly, we separate the input image into R, G and B channels. Secondly, we use the same CA transformation rules to process each channel. For each pixel in each channel, the corresponding pixel value is converted to an 8-bit binary representation, and the CA rules are applied iteratively. Thirdly, the first four bits are updated via XOR operations and the remaining four bits are inverted bitwise. By repeating this process multiple times, the system achieves a chaotic effect. Finally, the transformed R, G and B channels are merged back together, resulting in the overall encrypted output image.

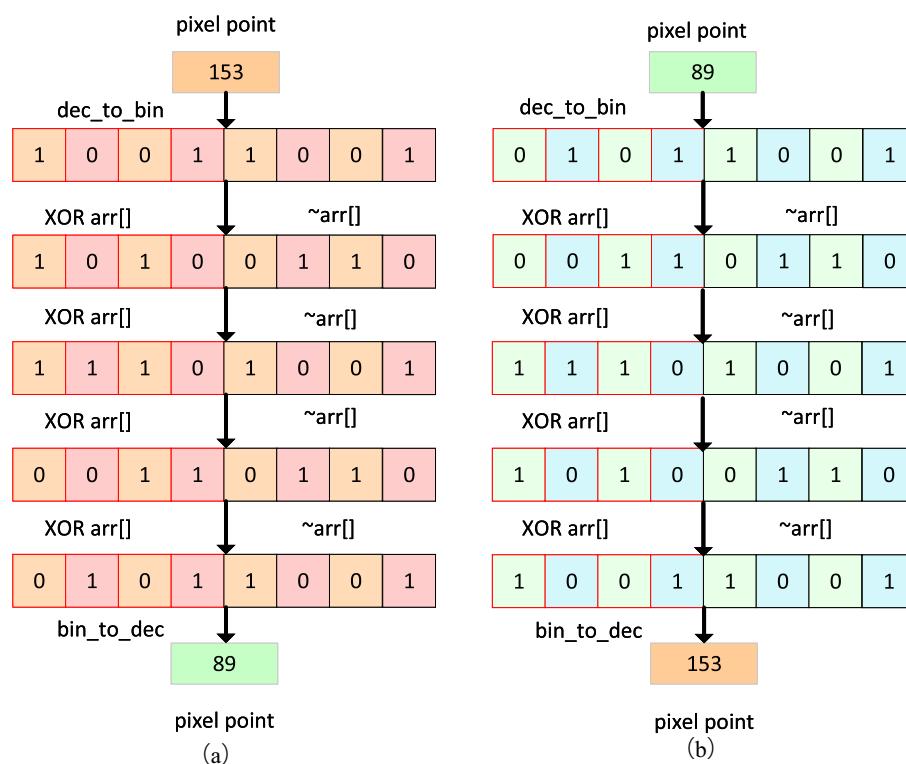


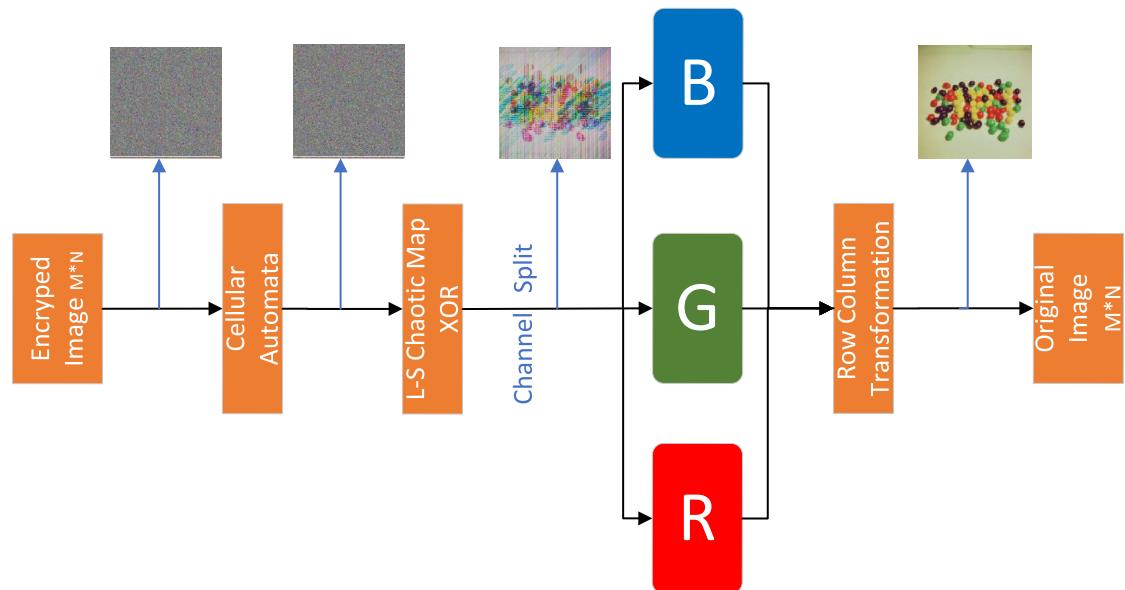
Fig. 6. Cellular Automata image encryption and decryption process. (a) is encryption process, (b) is decryption process.

Require: Input Image Matrix I , Output Matrix P
Ensure: Encrypted Image Matrix E

- 1: **Step 1: Split Image into Channels**
- 2: Split the input image I into three channels: B , G , and R
- 3: **Step 2: Cellular Automaton Transformation**
- 4: **for** each channel (B , G , R) **do**
- 5: **for** each pixel (i , j) in the channel **do**
- 6: Convert the decimal pixel value to an 8-bit binary array
- 7: **for** $k \leftarrow 1$ to 4 **do** ▷ Perform 4 iterations of binary updates
- 8: Apply XOR operation on the first 4 bits: $arr[k] \leftarrow arr[k] \oplus arr[k + 1]$
- 9: Apply NOT operation on the last 4 bits: $arr[k] \leftarrow \neg arr[k] \wedge 1$
- 10: **end for**
- 11: Convert the updated binary array back to a decimal pixel value
- 12: **end for**
- 13: **end for**
- 14: **Step 3: Merge Channels**
- 15: Merge the transformed B , G , and R channels into an encrypted image E
- 16: **return** E

Algorithm 2. Cellular Automaton Encryption**Decryption process**

The decryption process is the inverse of encryption, as shown in Fig. 7. The decryption process is implemented by Algorithm 3.

**Fig. 7.** Block diagram of the proposed image decryption.

Require: Encrypted Image E , Output Path P
Ensure: Decrypted Image D

- 1: **Step 1: Cellular Automaton Decryption**
- 2: **for** each channel (B, G, R) in E **do**
- 3: **for** each pixel (i, j) in channel **do**
- 4: Convert pixel value to binary
- 5: Apply XOR and NOT operations (4 iterations, reverse of encryption)
- 6: Convert back to decimal
- 7: **end for**
- 8: **end for**
- 9: Merge channels: $I_{\text{decell}} \leftarrow \text{Merge}(B, G, R)$
- 10: **Step 2: Chaotic Map Decryption**
- 11: Generate chaotic key K using logistic map (same key as encryption)
- 12: **for** each pixel (i, j) in I_{decell} **do**
- 13: $I_{\text{dechaotic}}[i, j] \leftarrow I_{\text{decell}}[i, j] \oplus K[z]$
- 14: **end for**
- 15: **Step 3: Pixel Position Restoration**
- 16: Split $I_{\text{dechaotic}}$ into channels: B, G, R
- 17: **for** each channel (B, G, R) **do**
- 18: Reverse circular shift on rows
- 19: Reverse row and element reversal
- 20: Reverse column interchange
- 21: Reverse column reversal
- 22: **end for**
- 23: Merge channels: $D \leftarrow \text{Merge}(B, G, R)$
- 24: **return** D

Algorithm 3. Decryption

Experimental results and analysis

This section analyzes the image encryption algorithm's performance, including histogram, pixel correlation, information entropy, key sensitivity, key space, robustness, and blocking attack analyses.

Encryption and decryption results

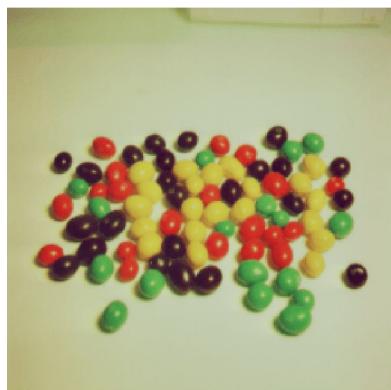
We evaluate the encryption performance on several standard images from the CVG-UGR and USC-SIPI datasets. Figure 8 shows the simulation results of three color images "Jelly", "House", "Baboon". The Jelly image, with a size of 256×256 pixels, are sourced from USC-SIPI. The House image, measuring 256×256 pixels, is obtained from USC-SIPI. The Baboon image, with a size of 512×512 pixels, are sourced from CVG-UGR. As illustrated in Fig. 8, the cipher image lacks any discernible resemblance to the original image, thereby demonstrating the efficacy of the image encryption algorithm in concealing image information. In addition, the hardware configurations utilized are Intel CPU RTX 4060 and 16 GB RAM, while the software configurations are Windows 11 and Pycharm 2023.

Histogram analysis

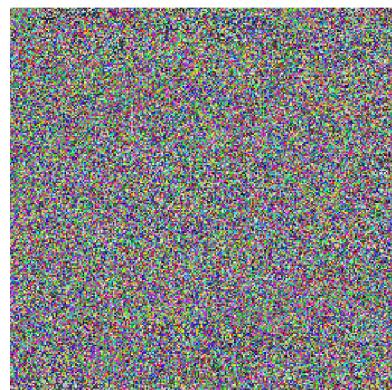
Histograms are important tools for qualitative analysis of digital images. After applying the encryption algorithm, the original image becomes visually unrecognizable. The encrypted image undergoes multi-layer row and column transformations, chaotic map, and Cellular automata encryption. This process results in a more uniform distribution of pixel values and reduced frequency fluctuations. Figure 9 shows the Jelly image. Figure 10 displays the encrypted image. The histogram of original's image, shown in the form of a curve Fig. 11, illustrates the distribution of pixel values in the plaintext image. The histogram clearly shows the difference between the plaintext and ciphertext images, with the plaintext image exhibiting high and low levels. The encrypted ciphertext image Fig. 12 exhibits random noise with a more uniform distribution in its histogram. The frequency change in encrypted pixel values is minimal. It indicates that the information of the original image is effectively hidden, thus guaranteeing the security of the image.

Correlation analysis

Correlation analysis can be used to evaluate the security and effectiveness of encryption algorithms for hyper chaotic systems. In ordinary images, the pixel values of neighboring pixels have a small range of variation and relatively concentrated distribution. It makes them susceptible to differential attacks. In contrast, the pixel distribution of encrypted cryptographic images is more uniform. Therefore, the correlation between



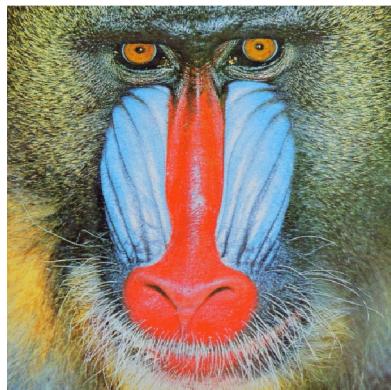
(a) Jelly



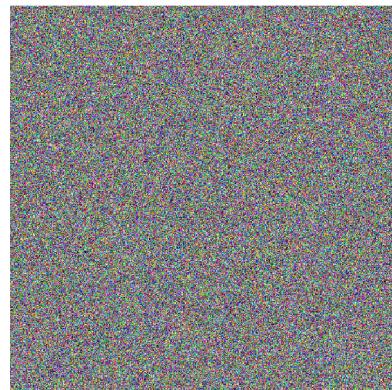
(b) Jelly



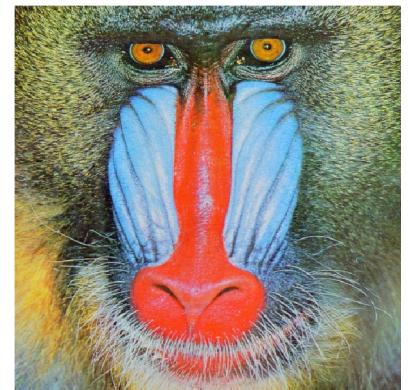
(c) Jelly



(d) Baboon



(e) Baboon



(f) Baboon



(g) House



(h) House



(i) House

Fig. 8. Encryption and decryption results of some pictures. (a), (d), (g) are original images. (b), (e), (h) are encrypted images. (c), (f), (i) are decrypt images.

neighboring pixels should be eliminated as much as possible in order to enhance the security of the image during the encryption process. In addition, the degree of correlation between pixels can be qualitatively analyzed by the correlation coefficient. Three thousand pairs of neighboring pixel points were selected from each of the plaintext image and ciphertext image for testing. Figure 13 shows the correlation test results between the selected pixel points and the neighboring pixel points. In the plaintext image, there is a strong correlation between the pixel points, indicated by a large correlation coefficient (see Table 2). Whereas in the encrypted image Fig. 14, the correlation between the pixel points is destroyed and the value of the correlation coefficient is close to zero. The correlation is very weak (see Table 3). The correlation coefficient can be calculated by Eq. 6.



Fig. 9. Original image.

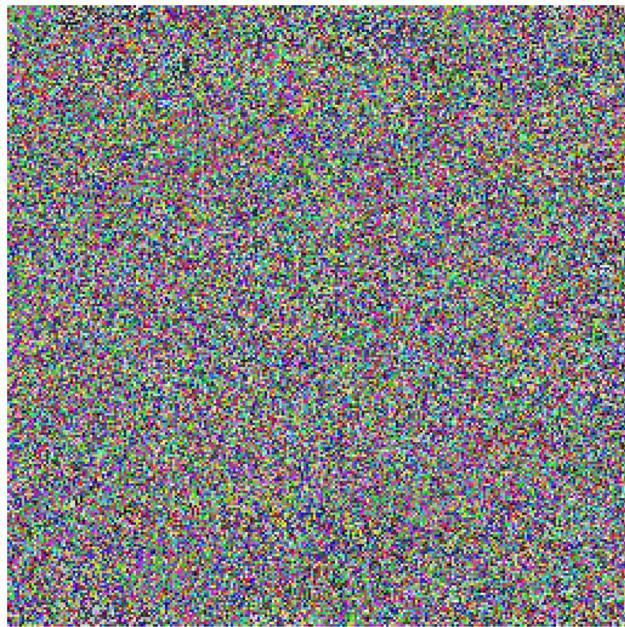


Fig. 10. Encryption image.

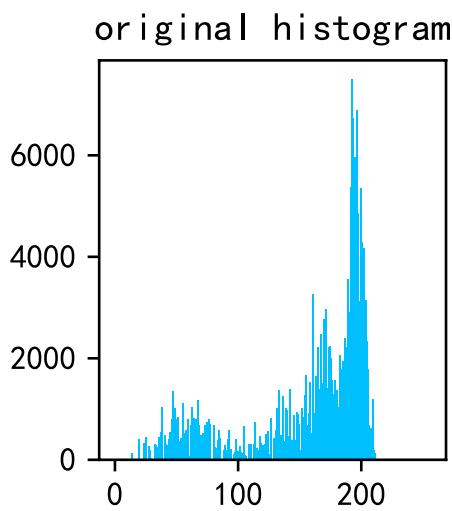


Fig. 11. Original image histogram.

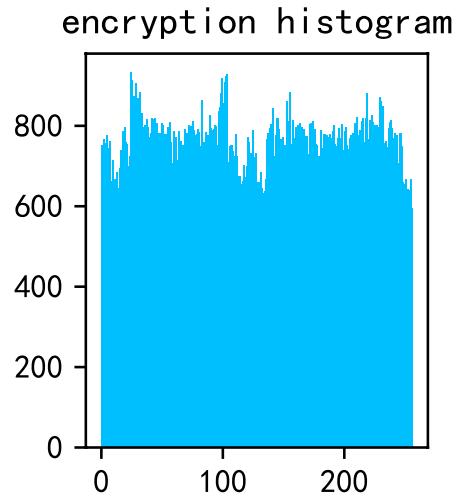


Fig. 12. Encryption histogram.

$$r_{xy} = \frac{\cot(x, y)}{\sqrt{D(x)} \sqrt{D(y)}} \quad (5)$$

$$\text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))(y_i - E(y)) \quad (6)$$

$$D(x) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))^2 \quad (7)$$

$$E(x) = \frac{1}{N} \sum_{i=1}^N x_i$$

Where x , y , and n represent the pixel values and number of pixel points of neighboring positions in horizontal, vertical, and diagonal directions. The correlation coefficient indicates the strength of the correlation, with larger absolute values indicating stronger correlations. Figure 10 shows that the pixel points of the plaintext image are concentrated near the line $y=x$ in all directions, indicating a small gap between neighboring pixel values. In contrast, the pixel points in the ciphertext image encrypted by the algorithm are uniformly distributed with a small correlation coefficient. The given situation implies that the ciphertext image conceals the information in the plaintext image more effectively. Table 2 presents the correlation coefficient results and comparison of the

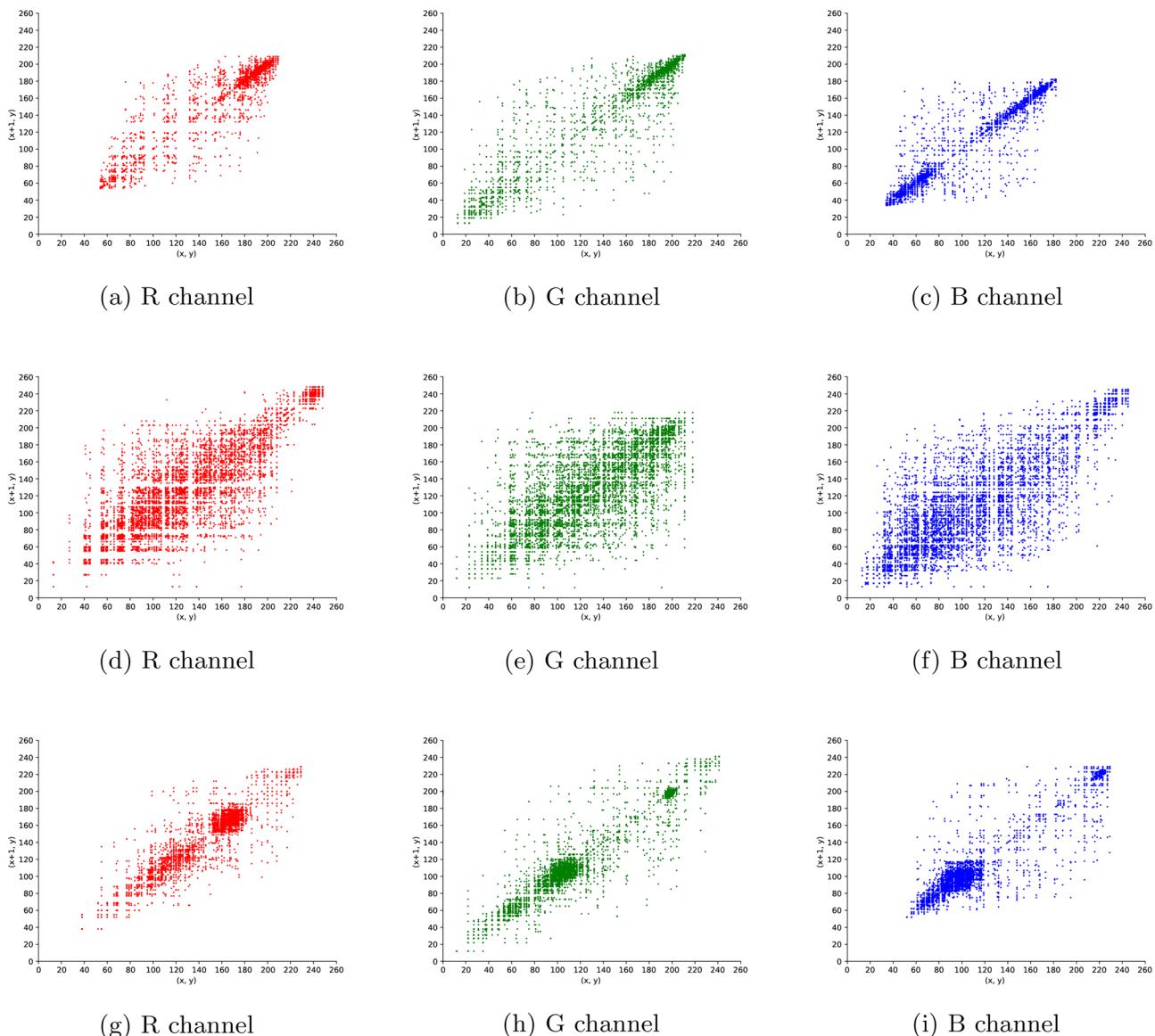


Fig. 13. Original image correlation, (a, c) is “Jelly”, (d, f) is “Baboon”, (g, i) is “House”.

| Images | | Jelly | Baboon | House |
|------------|---|--------|--------|--------|
| Horizontal | R | 0.9654 | 0.9213 | 0.9070 |
| | G | 0.9779 | 0.8694 | 0.9130 |
| | B | 0.9779 | 0.9028 | 0.9173 |
| Vertical | R | 0.9663 | 0.8561 | 0.9255 |
| | G | 0.9791 | 0.7812 | 0.9156 |
| | B | 0.9791 | 0.8684 | 0.9285 |
| Diagonal | R | 0.9573 | 0.8532 | 0.8540 |
| | G | 0.9622 | 0.7468 | 0.8425 |
| | B | 0.9622 | 0.8314 | 0.8683 |

Table 2. Correlation coefficients of plain images.

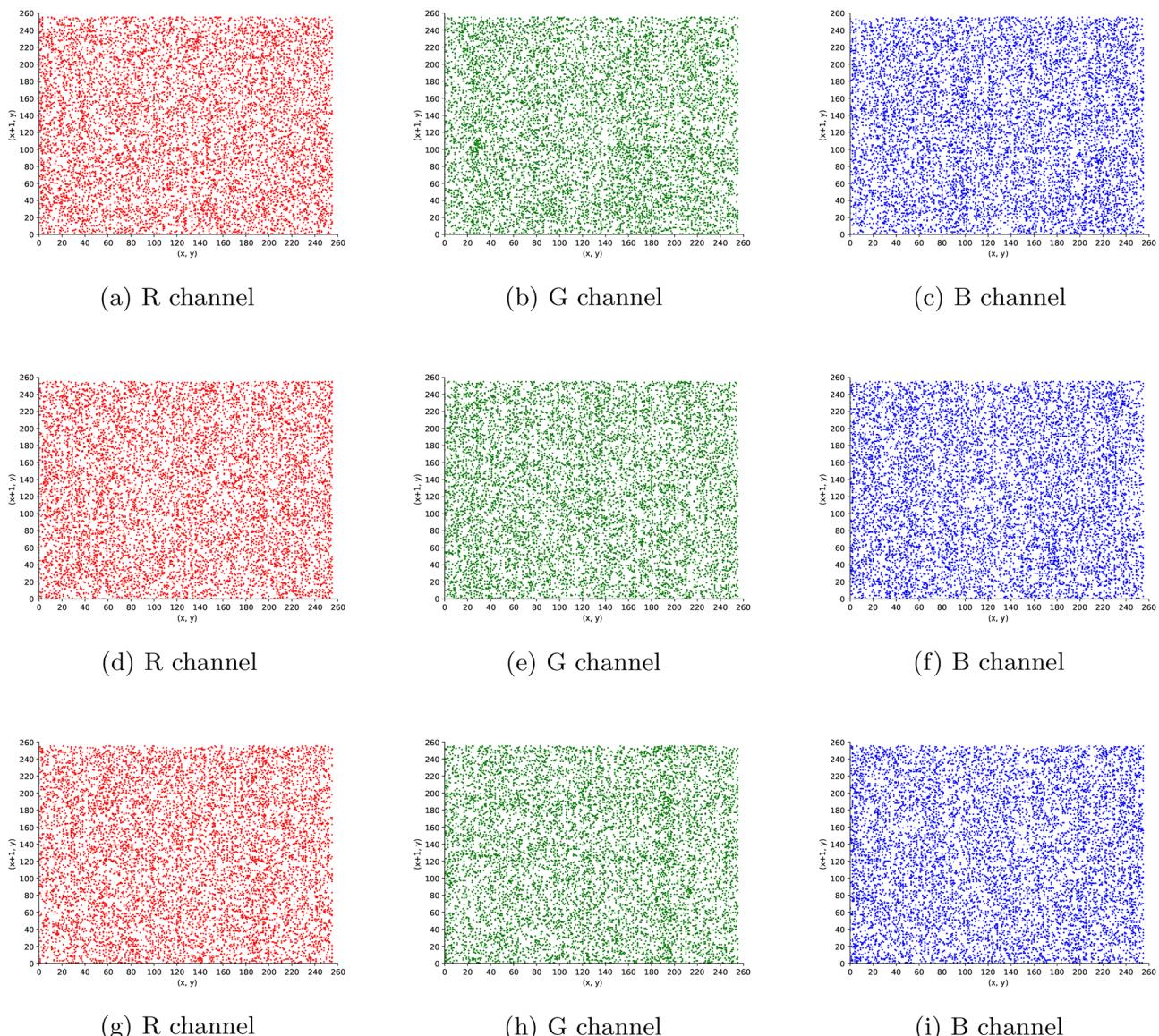


Fig. 14. encrypted image correlation, (a, c) is “Jelly”, (d, f) is “Baboon”, (g, i) is “House”.

| Images | | Jelly | Baboon | House | ³¹ | Ref. ³² (2025) | Ref. ³³ (2025) |
|------------|---|---------|---------|---------|---------------|---------------------------|---------------------------|
| Horizontal | R | -0.0188 | -0.0193 | 0.0221 | 0.0762 | -0.0099 | -0.0112 |
| | G | -0.0390 | 0.0113 | -0.0232 | 0.0875 | 0.0275 | -0.0112 |
| | B | -0.0261 | -0.0342 | 0.0020 | 0.0704 | 0.0085 | -0.0112 |
| Vertical | R | -0.0020 | -0.0327 | 0.0170 | 0.6654 | -0.0007 | -0.0115 |
| | G | -0.0569 | 0.0121 | -0.0114 | 0.0787 | -0.0099 | -0.0115 |
| | B | 0.0191 | 0.0060 | 0.0262 | 0.0598 | 0.0020 | -0.0115 |
| Diagonal | R | -0.0158 | -0.0348 | 0.0030 | 0.0671 | 0.0052 | 0.0173 |
| | G | 0.0016 | 0.0170 | 0.0175 | 0.0751 | 0.0154 | 0.0173 |
| | B | -0.0040 | 0.0254 | -0.0324 | 0.0637 | -0.0122 | 0.0173 |

Table 3. Correlation coefficient of encrypted image.

| | LSCM | ³⁴ | ³⁵ |
|---------|-------|---------------|---------------|
| entropy | 7.999 | 7.992 | 7.998 |

Table 4. Information entropy.

| Channel | NPCR | UACI | NPCR ³¹ | UACI ³¹ |
|---------|---------|---------|--------------------|--------------------|
| R | 99.5625 | 33.4481 | 99.60 | 33.39 |
| G | 99.5942 | 33.3751 | 99.61 | 33.40 |
| B | 99.6235 | 33.4237 | 99.59 | 33.37 |

Table 5. Key security analysis.

original plaintext images. Table 2 presents the correlation coefficient results and comparison of the encrypted images. The data in the tables indicates that the correlation coefficient results for the ciphertext image are close to 0, confirming that the algorithm exhibits very low correlation. The presentation shows the R, G, and B channels respectively. By comparing Tables 2 and 3, it can be observed that the correlation coefficient of neighboring pixels in the encrypted image is smaller than that reported in the literature³¹. It suggests that the algorithm's encrypted image offers higher security.

Information entropy

Information entropy is a metric used to quantify the uncertainty and confusion of an image. As the information entropy increases, the complexity and randomness of the image also increase. This leads to a reduction in the visual information of the image while enhancing its security. Eq. 8 as:

$$H = - \sum_{i=0}^L p(i) \log_2 p(i) \quad (8)$$

In Eq. 8, L represents the grey number level of the image, and $p(i)$ represents the probability of occurrence of grey value i . To ensure objectivity, subjective evaluations have been excluded. Upon analyzing the comparative literature in Table 4, it is evident that the mentioned chaotic algorithm encryption outperforms others in terms of information entropy. The algorithm exhibits values of information entropy that is closer to the theoretical value of information entropy 8 when compared to literature³⁴ and literature³⁵. This result confirms the algorithm's superiority in terms of randomness and demonstrates the cryptosystem's resistance to entropy attacks.

Key sensitivity analysis

Differential attacks generally exploit the ciphertext disparities resulting from minimal plaintext changes to launch targeted attacks²³. Ideally chaotic encryption should be key sensitive, which means that even a small change in a very weak value in the key should lead to a completely different encryption result. This property is called key sensitivity. In general, the key sensitivity of a chaotic cipher includes the sensitivity of the initial state of the chaotic map and the sensitivity of the control parameters. Two parameters are usually used to evaluate the key sensitivity: the number of pixel change rate (NPCR) and the uniform average change intensity (UACI). The closer the NPCR and UACI are to their optimal values, the more sensitive the encryption algorithm is to the security key, thereby increasing the algorithm's overall security. They are calculated as follows Eqs. (9) and (10):

$$\text{NPCR}(P_1, P_2) = \frac{\sum_{i=1}^M \sum_{j=1}^N |\text{Sign}(P_1(i, j) - P_2(i, j))| \times 100\%}{MN} \quad (9)$$

$$\text{Sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

$$\text{UACI}(P_1, P_2) = \frac{\sum_{i=1}^M \sum_{j=1}^N |P_1(i, j) - P_2(i, j)| \times 100\%}{255 - 0} \quad (10)$$

This paper presents the results of obtaining different encrypted images using a variable initial value x_0 with $x_0 + 1e-10$. The NPCR and UACI values for both images are shown in Table 5. The data indicates that the values obtained are close to the theoretical value, demonstrating that this chaotic encryption method is highly sensitive to the key. The difference between the two initial values x_0 is $1e-10$, as shown in Fig. 15 and Fig. 16. The encrypted images of the two images exhibit a significant difference, demonstrating sensitivity to the initial value.

By comparing the experiments, it is evident that our image encryption algorithm effectively enhances the security of the ciphertext image. This achieves the goal of privacy protection. The use of multilevel encryption makes our image encryption algorithm more difficult to decipher, and the matrix is more disordered, resulting in higher deciphering costs. Our algorithm outperforms those proposed in the literature. When compared to

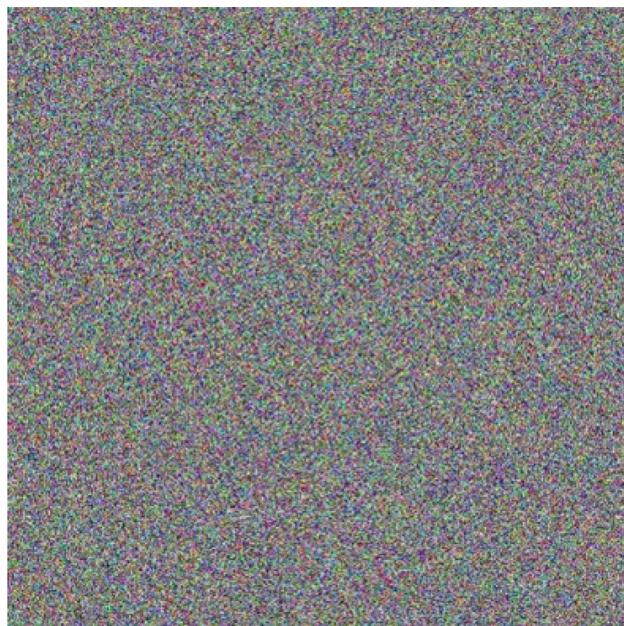


Fig. 15. This is the first encryption scheme represented as a x_0 encrypted image.

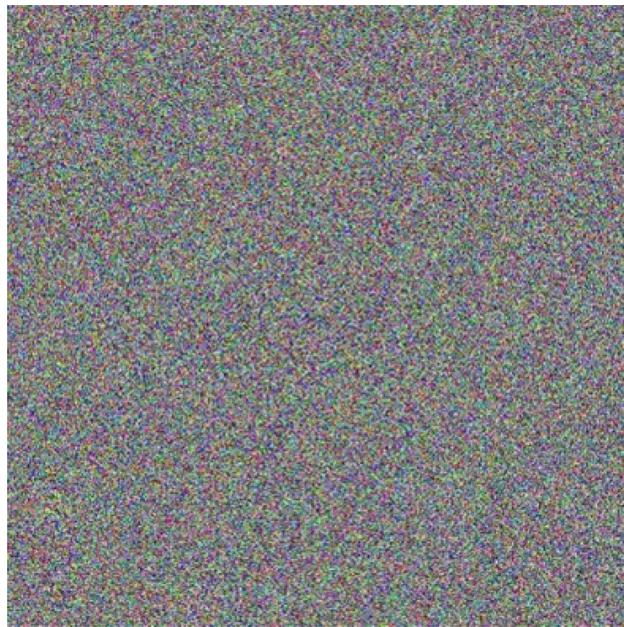


Fig. 16. This is the second encryption scheme represented as a $x_0+1e-10$ encrypted image.

other algorithms proposed in literature, this algorithm demonstrates high security and attack resistance due to its use of multilevel encryption.

Key space analysis

Key space is the range of key sizes that determine the security of an encryption algorithm. A larger key space provides better defense against exhaustive attacks, while a smaller key space is less likely to successfully defend against exhaustive attacks. When designing an encryption algorithm, it is important to ensure that the key space is as large as possible to improve security. In general, the key space should be at least 2^{100} . We take the key r, b, x and set the maximum precision to 15. The key space is approximated to be estimated as $10^{15} \times 10^{15} \times 10^{15} = 10^{45}$ $\approx 2^{150}$. The space in which our image encryption algorithm can generate chaotic sequences when computing the key space is 2^{150} . The key space of the encryption algorithm proposed in this paper is already greater than 2^{100} , and hence secure enough with enough key space to effectively avoid exhaustive attacks.

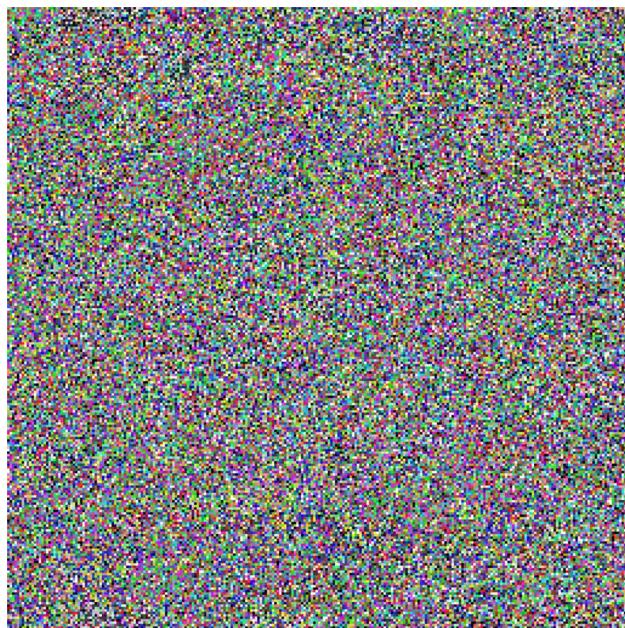


Fig. 17. salt noise image.



Fig. 18. salt noise decrypted image.

Robustness

It is conceivable that during the transmission or storage of encrypted images, their ciphertext pixels may be subject to loss or corruption. Consequently, a robust image encryption algorithm ought to possess the capability to withstand a certain degree of ciphertext pixel loss or corruption²³. During transmission, images are sometimes affected by various factors, such as noise. The noise may cause distortion or contamination of the image. These factors will cause the final decryption to fail if the image cannot be properly decrypted. This makes it more difficult to recover the ciphertext image. Therefore, the image encryption algorithm must be robust enough to resist noise attacks generated in real-world scenarios. In this paper, the problem encountered in real world scenarios is simulated by adding noise to the ciphertext image. The decryption effect is observed by decrypting the image with added noise. In this paper, we choose to use salt noise. The results of the experiments are shown in Fig. 17 salt noise image and Fig. 18 salt noise decryption image.

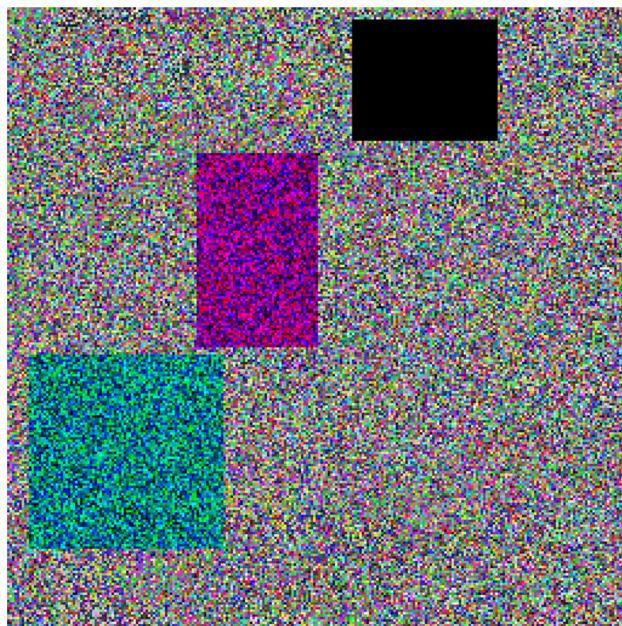


Fig. 19. Loss of encrypted image.

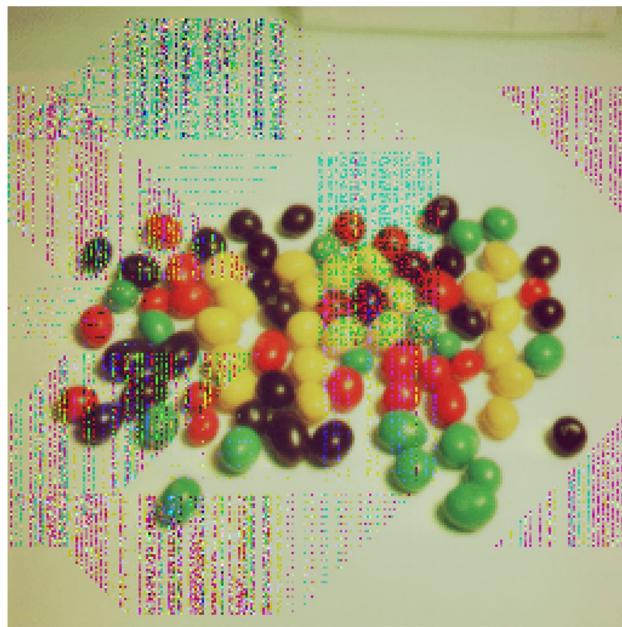


Fig. 20. Decrypted image.

Blocking attacks

During network communication, partial loss of images may occur, so the image encryption algorithm must be able to cope with this loss. After partial loss of data, the algorithm can still restore the encrypted image. To demonstrate the resistance of this algorithm to attacks, we modify the encrypted image by removing a pixel from the red channel, a pixel from the green channel, and a pixel from the blue channel. We then decrypt the modified image. If the original information is retained after decryption and the content of the original image can be visualized, it indicates that the proposed cipher is able to decrypt the lossy image. Figure 19 shows the removal (replaced by 0) of a pixel from the red channel of 80×80 pixels, a pixel from the green channel of 50×80 pixels, and a pixel from the blue channel of 60×50 pixels. The decrypted result is shown in Fig. 20.

| Time analysis | Image | Speed (s) |
|-----------------|--------------------|-----------|
| Proposed method | Jelly (256 × 256) | 0.52 |
| | House (256 × 256) | 0.41 |
| | Baboon (512 × 512) | 1.21 |
| ³⁶ | (512 × 512) | 0.0878 |
| ³⁶ | (256 × 256) | 0.0203 |
| ³⁷ | (512 × 512) | 1.7586 |
| ³⁷ | (256 × 256) | 0.4341 |

Table 6. Comparison of time analysis results.

Encryption efficiency

We list the encryption time for some images and compare it with some literature. Since we use multi-level encryption, some extra time will be spent in Table 6.

Conclusions

Visual data has been used as a focal point for privacy protection. This paper introduces a multi-level image encryption algorithm that combines matrix row and column transformation, chaotic map, and cellular automata to enhance the security and confidentiality of image data. Firstly, the original image is preprocessed by row and column transform, which enhances the encryption speed and the security of the image, and improves the antiattack ability. Secondly, a chaotic encryption method LSCM is introduced to make up for the periodicity defect of a single chaotic system and improve the security of the encryption system. Finally, the sequence is further obfuscated by cellular automata. It increases the complexity of the encryption system and makes it more difficult for the cracker to understand the operation of the encryption algorithm, thus improving the overall security of the encryption system. The algorithm has high security and attack resistance and is suitable for the security needs of protecting image data during network transmission.

In future work, a multi-layer application of encryption process is proposed in this algorithm. Nevertheless, there are still some limitations to be addressed. The time overhead of the algorithm may increase due to the inclusion of multiple layers, and further optimization will be pursued. The application of the algorithm to the practice of privacy protection will be explored with the aim of enhancing social benefits.

Data availability

The data presented in this study are openly available in CVG-UGR (<https://ccia.ugr.es/cvg/dbimagenes/>, and USC-SIPI (<https://sipi.usc.edu/database/>))

Received: 2 August 2024; Accepted: 29 May 2025

Published online: 01 July 2025

References

1. Adhikari, S. & Karforma, S. A novel audio encryption method using Henon-Tent chaotic pseudo random number sequence. *Int. J. Inf. Technol.* **13**(4), 1463–1471. <https://doi.org/10.1007/s41870-021-00714-x> (2021).
2. Ahuja, B., Doriya, R., Salunke, S., Hashmi, M. F. & Gupta, A. IoT-based multi-dimensional chaos mapping system for secure and fast transmission of visual data in smart cities. *IEEE Access* **11**, 104930–104945. <https://doi.org/10.1109/ACCESS.2023.3318014> (2023).
3. Alexan, W., Elkandoz, M., Mashaly, M., Azab, E. & Aboshousha, A. Color image encryption through chaos and KAA map. *IEEE Access* **11**, 11541–11554. <https://doi.org/10.1109/ACCESS.2023.3242311> (2023).
4. Alqhtani, M., Khader, M. M. & Saad, K. M. Numerical simulation for a high-dimensional chaotic Lorenz system based on Gegenbauer wavelet polynomials. *Mathematics* **11**(2), 472. <https://doi.org/10.3390/math11020472> (2023).
5. Arif, J. et al. A novel chaotic permutation-substitution image encryption scheme based on logistic map and random substitution. *IEEE Access* **10**, 12966–12982. <https://doi.org/10.1109/ACCESS.2022.3146792> (2022).
6. Chen, G., Mao, Y. & Chui, C. K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **21**(3), 749–761. <https://doi.org/10.1016/j.chaos.2003.12.022> (2004).
7. Dua, M., Wesanekar, A., Gupta, V., Bhola, M. & Dua, S. Differential evolution optimization of intertwining logistic map-DNA based image encryption technique. *J. Ambient Intell. Hum. Comput.* **11**(9), 3771–3786. <https://doi.org/10.1007/s12652-019-0158-0-z> (2020).
8. Elrefaey, A., Sarhan, A. & El-Shennawy, N. M. Parallel approaches to improve the speed of chaotic-maps-based encryption using GPU. *J. Real-Time Image Process.* **18**(6), 1897–1906. <https://doi.org/10.1007/s11554-020-01064-w> (2021).
9. Winarno, E., Nugroho, K., Adi, P. W. & Setiadi, D. R. I. M. Combined interleaved pattern to improve confusion-diffusion image encryption based on hyperchaotic system. *IEEE Access* **11**, 69005–69021. <https://doi.org/10.1109/ACCESS.2023.3285481> (2023).
10. Yaghouti Niyat, A., Moattar, M. H. & Niazi Torshziz, M. Color image encryption based on hybrid hyper-chaotic system and cellular automata. *Opt. Lasers Eng.* **90**, 225–237. <https://doi.org/10.1016/j.optlaseng.2016.10.019> (2017).
11. Yang, F., An, X. & Xiong, L. A new discrete chaotic map application in image encryption algorithm. *Phys. Scr.* **97**(3), 035202. <https://doi.org/10.1088/1402-4896/ac4fd0> (2022).
12. Yasser, I., Mohamed, M. A., Samra, A. S. & Khalifa, F. A chaotic-based encryption/decryption framework for secure multimedia communications. *Entropy* **22**(11), 1253. <https://doi.org/10.3390/e22111253> (2020).
13. Kumar, A. & Dua, M. Audio encryption using two chaotic map based dynamic diffusion and double DNA encoding. *Appl. Acoust.* **203**, 109196. <https://doi.org/10.1016/j.apacoust.2022.109196> (2023).
14. Li, C., Zhang, Y., Li, H. & Zhou, Y. Visual image encryption scheme based on inter-intra-block scrambling and weighted diffusion. *Vis. Comput.* **40**(2), 731–746. <https://doi.org/10.1007/s00371-023-02812-2> (2024).

15. Li, R., Liu, Q. & Liu, L. Novel image encryption algorithm based on improved logistic map. *IET Image Process.* **13**(1), 125–134. <https://doi.org/10.1049/iet-ipr.2018.5900> (2019).
16. Mansoor, S. & Parah, S. A. HAIE: A hybrid adaptive image encryption algorithm using Chaos and DNA computing. *Multimed. Tools Appl.* **82**(19), 28769–28796. <https://doi.org/10.1007/s11042-023-14542-7> (2023).
17. Feng, W., Zhang, J. & Qin, Z. A secure and efficient image transmission scheme based on two chaotic maps. *Complexity* **2021**(1), 1898998. <https://doi.org/10.1155/2021/1898998> (2021).
18. Martín, A., Hernández, A., Alazab, M., Jung, J. & Camacho, D. Evolving Generative Adversarial Networks to improve image steganography. *Expert Syst. Appl.* **222**, 119841. <https://doi.org/10.1016/j.eswa.2023.119841> (2023).
19. Peng, Y., Liu, J., He, S. & Sun, K. Discrete fracmemristor-based chaotic map by Grunwald–Letnikov difference and its circuit implementation. *Chaos Solitons Fractals* **171**, 113429. <https://doi.org/10.1016/j.chaos.2023.113429> (2023).
20. Ma, X., Wang, Z. & Wang, C. An image encryption algorithm based on tabu search and hyperchaos. *Int. J. Bifurc. Chaos* **34**(14), 2450170. <https://doi.org/10.1142/S0218127424501700> (2024).
21. Sheng, Y., Li, J., Di, X., Li, X. & Xu, R. An image encryption algorithm based on complex network scrambling and multi-directional diffusion. *Entropy* **24**(9), 1247. <https://doi.org/10.3390/e24091247> (2022).
22. Feng, W. et al. Exploiting robust quadratic polynomial hyperchaotic map and pixel fusion strategy for efficient image encryption. *Expert Syst. Appl.* **246**, 123190. <https://doi.org/10.1016/j.eswa.2024.123190> (2024).
23. Feng, W. et al. A novel multi-channel image encryption algorithm leveraging pixel reorganization and hyperchaotic maps. *Mathematics* **12**(24), 3917. <https://doi.org/10.3390/math12243917> (2024).
24. Song, X., Xu, D., Li, G. & Xu, W. Multi-image reorganization encryption based on s-l-f cascade chaos and bit scrambling. *J. Web Eng.* <https://doi.org/10.13052/jwe1540-9589.20410> (2021).
25. Feng, W., Qin, Z., Zhang, J. & Ahmad, M. Cryptanalysis and improvement of the image encryption scheme based on feistel network and dynamic DNA encoding. *IEEE Access* **9**, 145459–145470. <https://doi.org/10.1109/ACCESS.2021.3123571> (2021).
26. Wen, H. & Lin, Y. Cryptanalysis of an image encryption algorithm using quantum chaotic map and DNA coding. *Expert Syst. Appl.* **237**, 121514. <https://doi.org/10.1016/j.eswa.2023.121514> (2024).
27. Wen, H. & Lin, Y. Cryptanalyzing an image cipher using multiple chaos and DNA operations. *J. King Saud Univ. Comput. Inf. Sci.* **35**(7), 101612. <https://doi.org/10.1016/j.jksuci.2023.101612> (2023).
28. You, K., Chen, P. & Li, C. Cryptanalyzing an image encryption algorithm underpinned by a 3-D boolean convolution neural network **31**(4), 17–24 <https://doi.org/10.1109/MMUL.2024.3484414>
29. Wang, X., Li, Y. & Jin, J. A new one-dimensional chaotic system with applications in image encryption. *Chaos Solitons Fractals* **139**, 110102. <https://doi.org/10.1016/j.chaos.2020.110102> (2020).
30. Vidhya, R. & Brindha, M. A novel chaotic butterfly network topology based block scrambling and crown graph based bit-wise diffusion for image encryption. *Front. Comput. Sci.* **15**(2), 152804. <https://doi.org/10.1007/s11704-020-9196-8> (2021).
31. Jackson, J. & Perumal, R. A novel 2D hyperchaotic sine logistic map based image encryption scheme. *J. Opt.* <https://doi.org/10.1007/s12596-024-02280-4> (2024).
32. Xu, C. A novel color image encryption method using fibonacci transformation and chaotic systems. *EAI Endorsed Trans. Scalable Inf. Syst.* **12**(1), 1. <https://doi.org/10.4108/eetsis.5452> (2025).
33. Li, P., Xia, L., Fan, Y. & Qian, J. A novel six-wing chaotic system with line equilibrium and its application in image encryption. *Analog Integr. Circuits Signal Process.* **122**(2), 1–13 (2025).
34. Wang, R., Du, P., Zhong, W., Han, H. & Sun, H. Analyses and encryption implementation of a new chaotic system based on semitensor product. *Complexity* **2020**, 1–13. <https://doi.org/10.1155/2020/1230804> (2020).
35. Wang, T., Ge, B., Xia, C. & Dai, G. Multi-image encryption algorithm based on cascaded modulation chaotic system and block-scrambling-diffusion. *Entropy* **24**(8), 1053. <https://doi.org/10.3390/e24081053> (2022).
36. Li, H. et al. Exploiting dynamic vector-level operations and a 2d-enhanced logistic modular map for efficient chaotic image encryption. *Entropy* <https://doi.org/10.3390/e25081147> (2023).
37. Qian, K. et al. A novel image encryption scheme based on memristive chaotic system and combining bidirectional bit-level cyclic shift and dynamic dna-level diffusion. *Front. Phys.* <https://doi.org/10.3389/fphy.2022.963795> (2022).

Acknowledgements

This work is supported by the Key Research and Development Program of the Autonomous Region (Grant No.2022B01008), the National Key Research and Development Program of China (Grant No.2022ZD0115800), the Autonomous Region Science and Technology Program (No.2020A02001-1), the “Tianshan Talent” Research Project of Xinjiang (No. 2022TSYCLJ0037).

Author contributions

Shiji Sun: Software, Writing - original draft, Writing - review & edit, Visualization. Wenzhong Yang: Conceptualization, Funding acquisition, Supervision. Yabo Yin: Formal analysis, Writing - review & edit. Xiaodan Tian: Formal analysis, Supervision. Guanghan Li: Formal analysis. Xiangxin Deng: Formal analysis. All authors reviewed the manuscript.

Declarations

Competing Interests

The authors have no competing interests to declare that are relevant to the content of this article.

Ethical and informed consent for data used

We certify that the submitted work is original and has not been published elsewhere in any form or language (in whole or in part). The results of our experiments are presented in a clear and honest manner, without fabrication, falsification, or inappropriate data manipulation (including image-based manipulation).

Additional information

Correspondence and requests for materials should be addressed to W.Y. or Y.Y.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025