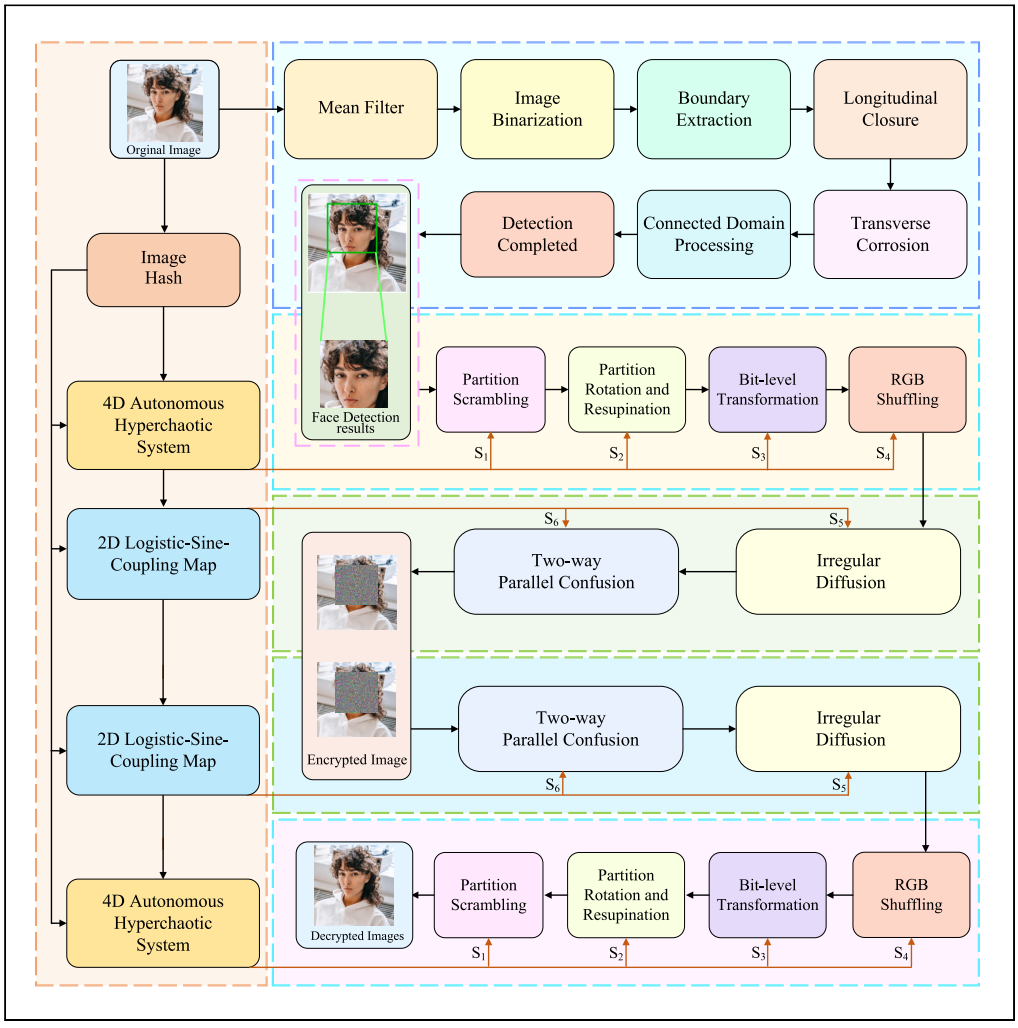
iScience

CellPress
OPEN ACCESS

Article

Face privacy protection scheme by security-enhanced encryption structure and nonlinear dynamics



Zhiyu Xie, Yiting Lin, Tengyu Liu, Heping Wen

wenheping@uestc.edu.com

Highlights

Facial recognition technology is used to identify facial images used to encrypt

Local encryption can improve the efficiency of encryption algorithms

The encryption algorithm driven by chaos sequences improves security

The proposed security enhancement structure can resist cryptographic attacks

Xie et al., iScience 27, 110768
September 20, 2024
https://doi.org/10.1016/j.isci.2024.110768

iScience

CellPress
OPEN ACCESS

Article

Face privacy protection scheme by security-enhanced encryption structure and nonlinear dynamics

Zhiyu Xie,[1,2] Yiting Lin,[1,2] Tengyu Liu,[1,2] and Heping Wen[1,2,3,*]

SUMMARY

This article proposes a secure communication enhancement scheme based on face detection and chaotic partition permutation. The scheme uses edge detection technology to detect facial information, which is then used as an encryption object. The hash value of the plaintext image is extracted as the secret key to the chaotic sequence generated by the chaotic system. Then a series of encryption operations are performed on the face image to obtain the final ciphertext image. In this article, two chaotic systems are used to generate pseudo-random chaotic sequences for different encryption steps. The initial key is computed by combining the hash function of the image and external parameters. The experimental results and security analysis show that the algorithm has excellent encryption effectiveness and security performance against various typical attacks.

INTRODUCTION

With the rapid development of network technology, massive image information is rapidly generated and shared through the network.[1–3] At the same time, unauthorized access and misuse issues have emerged. In recent years, encryption technology has received significant attention as an important privacy protection technique.[4–6] Current image encryption technologies[7–9] are typically combined with other techniques to enhance encryption security, such as chaos theory,[10–12] biometric encoding,[13–15] Fourier transform,[16–18] and so on.[19–21] Most existing image encryption schemes primarily focus on encrypting the entire image.[22–24] By fully utilizing the unpredictability and sensitivity of chaos, image encryption algorithms can provide strong protection against potential attacks or unauthorized access, making them the preferred solution in the field of image security.[25–27]

Looking at the current international situation, research on chaotic digital image encryption[28–30] has become a hot topic. With the popularization of multimedia information, local encryption schemes for specific sensitive information have gradually become the current mainstream security communication solution.[31,32] In recent years, many scholars have devoted themselves to research in this field and have achieved good results. In 2022, Gao et al.[33] introduced a new method for encrypting facial images. This method only encrypts the face region in the image, and the remaining regions remain unchanged. The experimental results show that the method has high computational efficiency and strong security. In 2023, Du et al.[34] proposed a new multi-face image encryption scheme based on non-adjacent dynamic coupling mapping lattice, confirming that the algorithm has high security. In 2024, Li et al.[35] proposed a ciphertext face recognition system based on secure inner product protocols and analyzed its correctness, security, and time expenditure, all of which yielded good results. From these studies, it can be seen that most secure communication algorithms based on face recognition have achieved satisfactory results, significantly advancing the development of information security technology and leading to the emergence of more encryption algorithms. Although these studies have yielded satisfactory results, as encryption systems become increasingly complex and data volumes grow exponentially, reducing data storage space and transmission bandwidth is becoming increasingly crucial. Therefore, proposing a face image encryption algorithm constructed using chaotic mappings becomes an especially important and urgent task to defend against various illegal attacks.

In response to the above problem, this article proposes a scheme to protect the privacy of portrait information in images using chaotic partition scrambling technology. First, the original image containing faces is subjected to face detection using edge detection recognition technology, and the detection results are used for further encryption. Then, the feature values of the original image are processed to obtain the key for generating the chaotic system and the chaotic sequence required for encryption. By using the generated chaotic sequence for partition permutation, irregular diffusion, and bidirectional parallel confusion, the final ciphertext is obtained. The experimental results show that the algorithm has excellent encryption effect and good encryption efficiency, and the proposed image encryption algorithm can securely resist various attacks.

[1]University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan 528402, China
[2]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[3]Lead contact
*Correspondence: wenheping@uestc.edu.com
https://doi.org/10.1016/j.isci.2024.110768

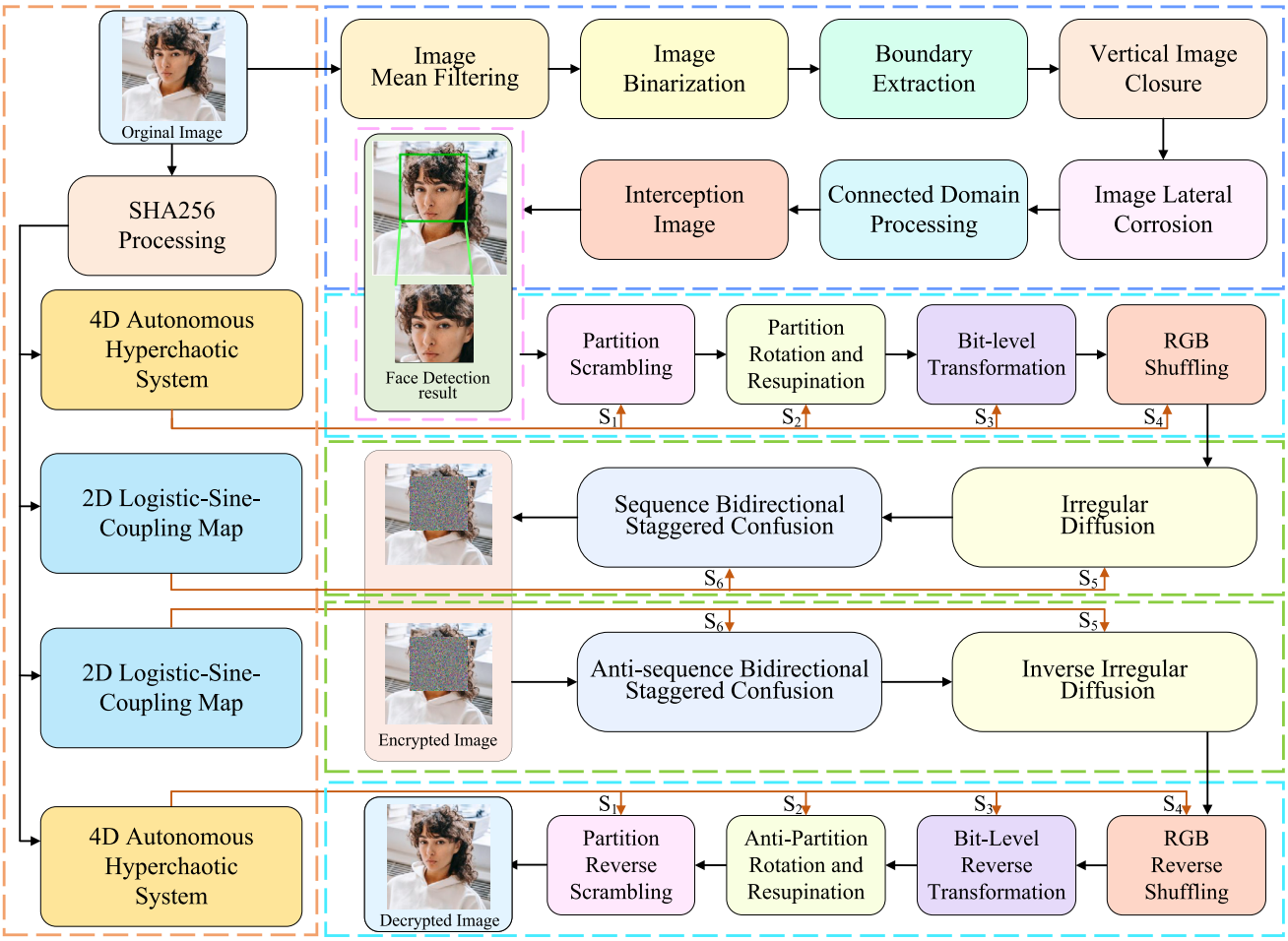


Figure 1. The proposed algorithm flowchart

The main innovations and contributions of this article are as follows:

(1) Most existing image encryption schemes mainly focus on encrypting the entire image. This often leads to loss of encryption performance due to processing redundant information. In order to enhance the security and efficiency of the encryption system, this algorithm encrypts facial images. In this article, facial recognition technology is used to encrypt facial images with a focus, without considering protecting redundant information. Encrypting facial images locally enhances the security and efficiency of the encryption system.

(2) Traditional encryption algorithm is relatively simple compared with traditional permutation algorithms and encoding algorithms. The algorithm dynamically adjusts the permutation algorithm based on chaotic sequence to achieve excellent encryption effect in the shortest possible time. The final ciphertext is obtained by using irregular diffusion and sequence bidirectional staggered confusion.

Figure 2. Image RGB conversion

Figure 3. Image mean filtering

Experimental results show that the algorithm effectively enhances the security of image encryption and improves the security and robustness of the encryption algorithm, making it more resistant to attacks and ensuring the confidentiality of transmitted images.

(3) Many existing encryption algorithm structures are not reasonable. If there is no relevant plaintext or ciphertext feedback, they are easily susceptible to known-plaintext or chosen-plaintext attacks. This secure image encryption scheme uses a dynamic feedback mechanism to update encryption keys based on encrypted data. Building upon our current foundation of cryptanalysis research,[36–38] it enhances security and the ability to resist attacks such as chosen-plaintext and chosen-ciphertext attacks.

The rest of the article is organized as follows. Section related works contains the related works. Section the proposed face image privacy protection scheme discusses the encryption algorithm. Section results and analysis presents the results and analysis. The last section concludes the article.

RELATED WORKS

The adopted chaotic system

Chaotic systems have properties such as strong ergodicity, randomness, and output uncertainty. Integrating chaotic systems into encryption schemes can enhance security by expanding the key space and improving computational efficiency. Using multiple chaotic systems further introduces complex nonlinear characteristics, thereby increasing the complexity and challenge for potential attackers.

4D-AHS

Applying controllers $\lambda_1 x_4, \lambda_2 x_4, \lambda_3 x_4$ to the state equation of 3D chaotic system, $\Upsilon$and setting $x_4 = -\tau x_1$, the following adaptive 4D autonomous hyperchaotic system[39] used in this article can be constructed.

$$\begin{cases} x_1 = \alpha(x_2 - x_1) + \lambda_1 x_4 \\ x_2 = \beta x_1 - x_1 x_3 + \lambda_2 x_4 \\ x_3 = -\Upsilon x_3 + x_1 x_2 + \lambda_3 x_4 \\ x_4 = -\tau x_1 \end{cases} \quad \text{(Equation 1)}$$

where $\alpha, \beta, \Upsilon, \tau, \lambda_1, \lambda_2, \lambda_3$ are the control parameters of the system. When $\alpha = 35, \beta = 35, \Upsilon = 3, \tau = 5, \lambda_1 = 1, \lambda_2 = 0.2$ and $\lambda_3 = 0.3$, the system exhibits hyperchaotic behavior.

Figure 4. Image binarization

Figure 5. Morphological boundary extraction

2D-LSCM

The security and efficiency of chaotic image encryption depend heavily on the properties of the chaotic sequence. In this study, a 2D discrete chaotic system[40] is chosen for image encryption due to its simplicity, ease of implementation and superior performance. The definition of the system is as follows:

$$\begin{cases} x_i = \sin(\pi(4rx_{i-1}(1 - x_{i-1}) + (1 - r)\sin(\pi y_{i-1}))) \\ y_i = \sin(\pi(4ry_{i-1}(1 - y_{i-1}) + (1 - r)\sin(\pi x_i))) \end{cases} \quad \text{(Equation 2)}$$

where r is the system control parameter.

SHA256

The SHA256 algorithm is a one-way hash function used in cryptography and information security. Its primary purpose is to transform messages of any length into shorter, fixed-length hash values to ensure data integrity and security. This algorithm is widely used in data integrity verification, password security, and digital signatures. Unlike other encryption schemes, the SHA256 algorithm does not use a single hash value as a key.

THE PROPOSED FACE IMAGE PRIVACY PROTECTION SCHEME

This article uses two chaotic systems and the SHA256 hash function to generate distinct chaotic keys, as opposed to traditional algorithms. The 4D-AHS is used for chaos-driven three-dimensional partition permutation, while the 2D-LSCM generates the chaotic sequence for diffusion and confusion. This improves the algorithm's security and makes code cracking more difficult. The flowchart of algorithm design is shown in Figure 1.

Face recognition

To begin the face detection process, the RGB image must first be converted to grayscale for processing, as demonstrated in Figure 2.

Next, the image is subjected to mean filtering operations to weaken the detailed parts in the image, avoiding the impact of jagged edges and redundant information on the formation of later connected domains. As can be seen in Figure 3, mean filtering reduces information redundancy in the image.

As shown in Figure 4, the image is binarized to reduce the computational load of the subsequent morphological operations.

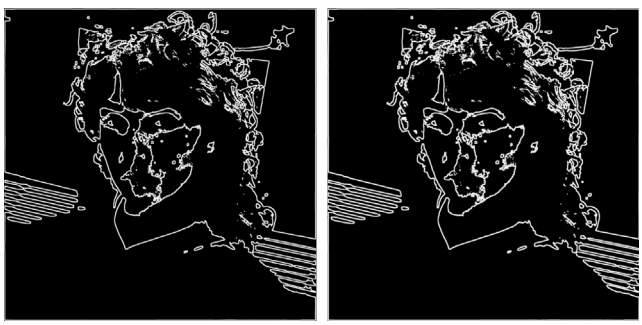


Figure 6. Border bolding

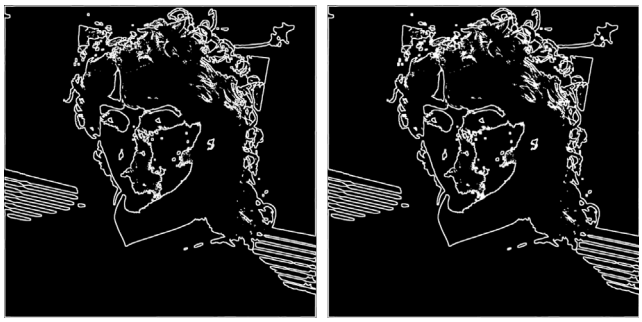


Figure 7. Image hole filling

Meanwhile, considering the diversity of skin colors, it is difficult to form large connected regions in the facial region after binarizing images of people with darker skin colors. To address this issue, this study adopts a morphological boundary extraction method. By using a sufficiently large structuring element, large closed connected domains can be formed. As can be seen in Figure 5, the boundaries are successfully extracted, forming a relatively large connected domain in the face region.

In addition, Figures 6 and 7 show that after thickening the boundaries and filling the holes, the black blocks in the eye area are eliminated, thus enlarging the facial connected domain.

As shown in Figure 8, this step uses vertically elongated structural elements for vertical closure operations. Because elements such as the face, neck, hair, and clothing have vertical distribution characteristics, it is easy to separate these adjacent elements when performing morphological boundary extraction, which adversely affects the determination of connected domains. Therefore, in this study, vertically elongated structural elements are used for vertical closure operations to reconnect the upper and lower regions of the face.

Next, use horizontally elongated structural elements for horizontal erosion operations. This is because there are many connected domains below the head region which can easily interfere with the determination of the largest connected domain. As this part is mostly vertically distributed, horizontal erosion can segment these large connected domains. It is important to note that the degree of segmentation should not be too large to ensure that the previous closure operation remains meaningful. Figure 9 shows that after the vertical closure operation, the connected domains are connected and the facial connected domain continues to expand. Subsequently, the horizontal erosion reduces the connected domains at the bottom of the image while preserving the size of the facial connected domain as much as possible.

After a number of operations, it is found that factors such as the background clutter in Figure 10 can also generate numerous connected domains that may interfere with the final judgment and therefore need to be eliminated.

After several rounds of screening, the largest connected region among the remaining regions is selected and surrounded by a rectangular box as a result of face detection, as shown in Figure 11.

The final face detection image is shown in Figure 12.

Chaotic initial value disturbance and sequence preprocessing

After performing feature extraction on the plaintext image P using the SHA256 function, a fixed-length character-based 64-bit hexadecimal number is obtained. To facilitate computation, each digit is converted to a decimal number. Each digit is represented by the variable $Hash$, respectively, denoted as $Hash_1$, $Hash_2$, $Hash_3 \cdots Hash_{64}$.

The image hashes are processed mathematically to obtain the initial key parameters. The hash values of the images, represented by variables a and b, are used as the initial values for 2D-LSCM. Variables c and d are used as control parameters for 2D-LSCM and iteration counts for 4D-AHS, respectively, after further processing by a and b.

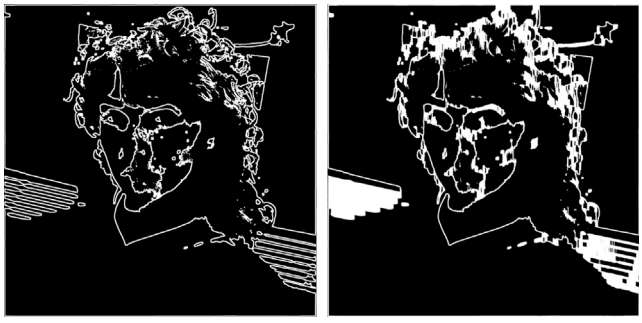


Figure 8. Vertical image closure

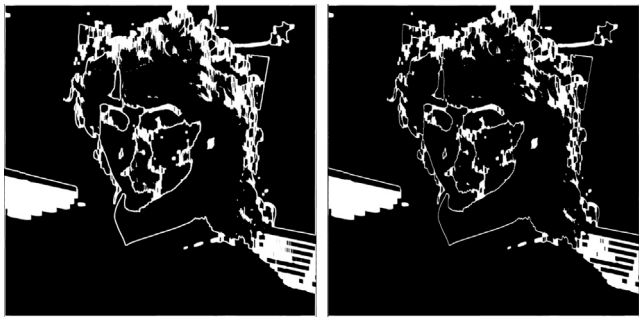


Figure 9. Image lateral corrosion

However, it is important to note that in order to further encrypt the key, 2D-LSCM operates as follows:

$$\begin{cases} R_1 = 2D\_LSCM(a+0.1, b+0.1, c, 3\times H\times W) \\ R_2 = 2D\_LSCM(a+0.2, b+0.2, c, 3\times H\times W) \end{cases} \quad \text{(Equation 3)}$$

where H represents the height of the image and W represents the width of the image; $3\times H\times W$ represents the length of the generated chaotic sequence; R_1 and R_2 are the initial chaotic sequences generated by the chaotic system through iteration. Then, further processing is performed on the obtained chaotic sequence following the steps below:

$$\begin{cases} S_j = (R_1 \times 10000) \bmod 255 \\ S_k = (R_2 \times 10000) \bmod 255 \end{cases} \quad \text{(Equation 4)}$$

where S_j and S_k are used in irregular diffusion and bidirectional parallel confusion, respectively.

Chaos-driven three-dimensional partition permutation

Since permutation in existing encryption schemes only changes the positions of pixel values without changing the pixel values themselves, and does not process the color components of the RGB layers, this article proposes chaos-driven three-dimensional partition permutation based on chaotic sequence control that changes both the positions and values of pixels to achieve security enhancement. The specific operational flowchart is shown in Figure 13.

Step1: First, the plaintext image *Image* with a size of $H\times W$ is partitioned and converted into matrices with a number of rows and columns that is a multiple of 8. If the matrix does not have enough elements, zero padding is performed. The following steps outline the specific operations:

$$P(x, y, z) = Image \quad \text{(Equation 5)}$$

where x is $W + W_0$; y is $H + H_0$; W_0 and H_0 denote the number of '0' to be added and z represents the dimensions of image I.

Step2: Preprocess P from a 3D array to a 5D array, the specific operations are as follows:

$$Pa(x, y, z, i, j) = P((i-1)\times Px + 1, (j-1)\times Px + 1 : j \times Px, z) \quad \text{(Equation 6)}$$

where x, y, and z are the rows, columns, and dimensions of a 5D array, respectively. i and j are the specific positions after blocking, $i, j = 1, 2, \ldots$ and Px is the number of unit blocks after partition.

Step3: Read the hash features of the image and compute the obtained key d, and after iterating the key through a chaotic system, obtain four chaotic sequences S_1, S_2, S_3 and S_4. Use mathematical methods to process and obtain the sequences S_1', S_2', S_3' and S_4', as follows:

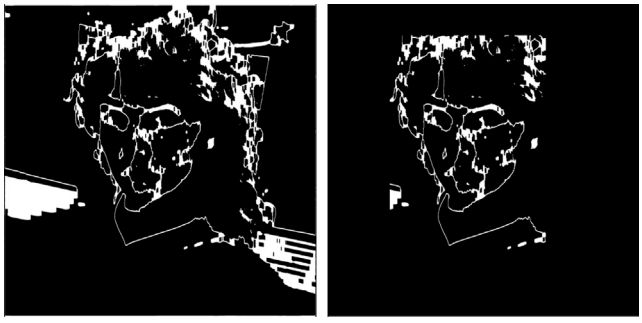


Figure 10. Eliminate redundant connected domain of image

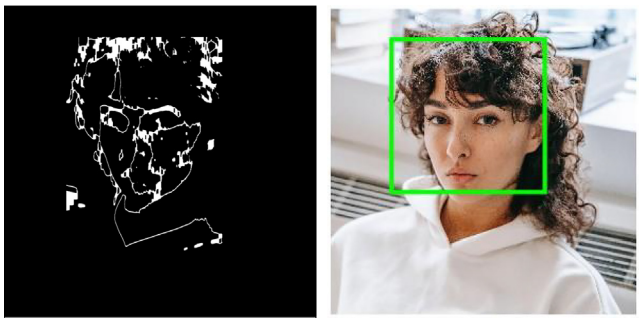


Figure 11. Maximum connectivity domain of intercepted image

$$\begin{cases} S_1' = \left(S_1 \times 10^{10}\right) \bmod (H \times W)/L^2 \\ S_2' = \left(S_2 \times 10^{10}\right) \bmod 6 \\ S_3' = \left(S_3 \times 10^{10}\right) \bmod 2 \\ S_4' = \left(S_4 \times 10^{10}\right) \bmod 6 \end{cases} \quad \text{(Equation 7)}$$

where L is the size of each block after partitioning the image.

Step4: As shown in Figure 14, the sequence S_1' obtained in Step 3 is used to scramble the partitioned image P.

$$\begin{cases} t = P(x, y, z, i) \\ P(x, y, z, i) = P(x, y, z, S_1'(i)) \\ P(x, y, z, S_1(i)) = t \end{cases} \quad \text{(Equation 8)}$$

where $x = [1,2\cdots H]$, $y = [1,2\cdots W]$, $i = 1,2\cdots\frac{H\times W}{(Px)^2}$, z is the dimension of the matrix, and t is an intermediate variable. The matrix after partitioning is called Pa.

Step5: Based on the sequence S_2', perform rotation encryption on the initially scrambled image to obtain the post-rotation encryption image matrix. Figure 15 shows the operation diagram, and the specific operation formula is as follows:

- When $S_2' = 1$ rotate the image 90° clockwise;

$$\begin{cases} Pa(k, l, z, i, j) = Pa(Px + 1 - l, k, z, i, j) \\ Pa(Px + 1 - l, k, z, i, j) = Pa(Px + 1 - k, Px + 1 - l, z, i, j) \\ Pa(Px + 1 - k, Px + 1 - l, z, i, j) = Pa(1, Px + 1 - k, z, i, j) \\ Pa(l, Px + 1 - k, z, i, j) = Pa(k, l, z, i, j) \end{cases} \quad \text{(Equation 9)}$$

- When $S_2' = 2$ rotate the image 180° clockwise;

$$\begin{cases} Pa(k, l, z, i, j) = Pa(Px + 1, Px + 1 - l, z, i, j) \\ Pa(Px + 1 - k, Px + 1 - l, z, i, j) = Pa(k, l, z, i, j) \\ Pa(k, l + Px, z, i, j) = Pa(Px + 1 - k, Px/2 + 1 - l, z, i, j) \\ Pa(Px + 1 - k, Px/2 + 1 - l, z, i, j) = Pa(k, l + Px, z, i, j) \end{cases} \quad \text{(Equation 10)}$$

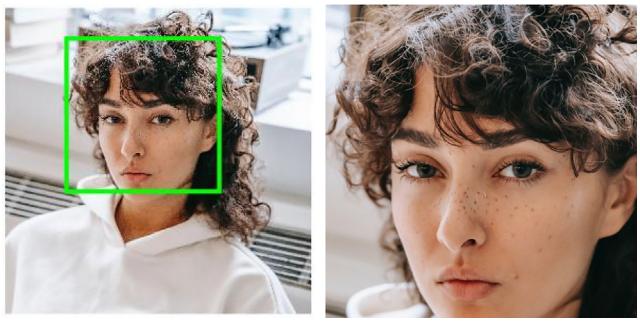


Figure 12. Face detection effect

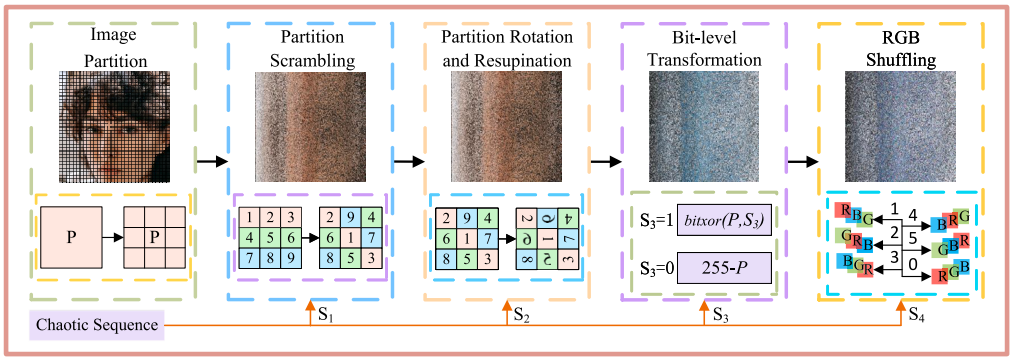


Figure 13. Chaos-driven three-dimensional partition permutation flowchart

- When $S_2' = 3$ rotate the image 270° clockwise;

$$\begin{cases} Pa(k,l,z,i,j) = Pa(Px+1, Px+1-l, z, i, j) \\ Pa(Px+1-k, Px+1-l, z, i, j) = Pa(k,l,z,i,j) \\ Pa(k, l+Px, z, i, j) = Pa(Px+1-k, Px/2+1-l, z, i, j) \\ Pa(Px+1-k, Px/2+1-l, z, i, j) = Pa(k, l+Px, z, i, j) \end{cases} \quad \text{(Equation 11)}$$

- When $S_2' = 4$ flip the image horizontally;

$$\begin{cases} Pa(k,l,z,i,j) = Pa(k, Px+1-l, z, i, j) \\ Pa(k, Px+1-l, z, i, j) = Pa(k,l,z,i,j) \\ Pa(Px/2+k, l, z, i, j) = Pa(Px/2+k, Px+1-l, z, i, j) \\ Pa(Px/2+k, Px+1-l, z, i, j) = Pa(Px/2+k, l, z, i, j) \end{cases} \quad \text{(Equation 12)}$$

- When $S_2' = 5$ flip the image vertically;

$$\begin{cases} Pa(k,l,z,i,j) = Pa(Px+1-k, l, z, i, j) \\ Pa(Px+1-k, l, z, i, j) = Pa(k,l,z,i,j) \\ Pa(k, Px/2+l, z, i, j) = Pa(Px+1-k, Px/2+l, z, i, j) \\ Pa(Px+1-k, Px/2+l, z, i, j) = Pa(k, Px/2+l, z, i, j) \end{cases} \quad \text{(Equation 13)}$$

where k, l are control parameters with specific values of $1,2,\ldots,Px/2$. z is the dimension of the array. i , j are variables with specific values ranging from $1,2,\ldots,m/Px$; $1,2,\ldots,n/Px$.

Step6: Based on the sequence S_3', if the random sequence $S_3' = 1$, perform bitwise expansion on the image matrix after rotation encryption, and perform bit-level transformation with the chaotic sequence; if the random sequence $S_3' = 0$, take the difference of 255. The specific formula is:

$$\begin{cases} Pa(x,y,z,i,j) = 255 - Pa(x,y,z,i,j) & if \quad S_3' = 0 \\ Pa(x,y,z,i,j) = bitxor(Pa(x,y,z,i,j), 1) & if \quad S_3' = 1 \end{cases} \quad \text{(Equation 14)}$$

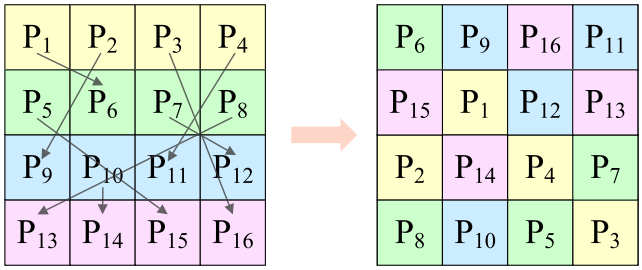


Figure 14. Partition scrambling flowchart

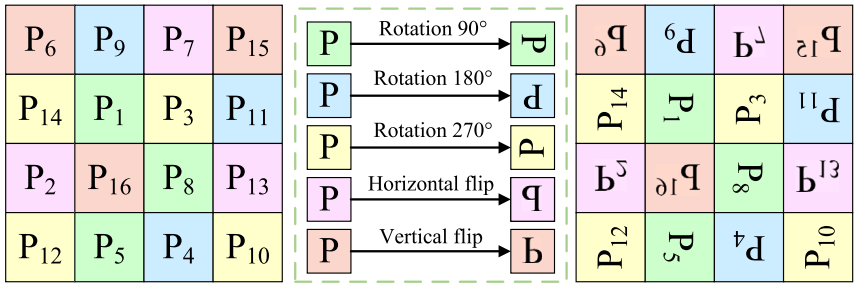


Figure 15. Partition rotation and resupination flowchart

Step7: Select the appropriate RGB shuffling based on the random sequence S_4', as follows:

- When $S_4' = 1$, the three-dimensional components of color are transformed from *RGB* to *RBG*. The G color channel is swapped with the B color channel and the data transformation is performed using the ratio specificity operation. The specific operation is as follows:

$$\begin{cases} Pa(x,y,2,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,3,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,2,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \end{cases} \quad \text{(Equation 15)}$$

- When $S_4' = 2$, the three-dimensional components of color are transformed from *RGB* to *GRB*. The G color channel is swapped with the R color channel and the data transformation is performed using the ratio specificity operation. The specific operation is as follows:

$$\begin{cases} Pa(x,y,1,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \\ Pa(x,y,2,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \\ Pa(x,y,1,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \end{cases} \quad \text{(Equation 16)}$$

- When $S_4' = 3$, the three-dimensional components of color are transformed from *RGB* to *BGR*. The B color channel is swapped with the R color channel and the data transformation is performed using the ratio specificity operation as follows:

$$\begin{cases} Pa(x,y,1,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \\ Pa(x,y,3,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \\ Pa(x,y,1,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \end{cases} \quad \text{(Equation 17)}$$

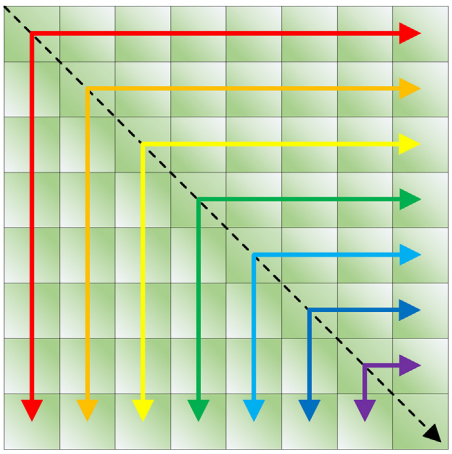


Figure 16. Sequence bidirectional staggered confusion sketch

Algorithm 1. Sequence bidirectional staggered confusion

Input: The chaotic matrix S reshaped by the chaotic sequence S_k, intermediate ciphertext image C_i

Output: Final ciphertext image C

1: $[M, N] = \text{size}(Ci)$;
2: **for** $i \leftarrow 1$ to M **do**
3: **for** $ci \leftarrow i+1$ to N **do**
4: $I(i, ci) = \text{mod}(IC(i, ci) + EI(i, ci-1), 256)$;
5: **end for**
6: **for** $ri \leftarrow i+1$ to N **do**
7: $I(ri, i) = \text{mod}(IC(ri, i) + EI(ri-1, i), 256)$;
8: **end for**
9: **end for**
10: $C = \text{mod}(I + S, 256)$;

- when $S_4' = 4$, the three-dimensional components of color are transformed from *RGB* to *BGR* and then to *BRG*. Following the alignment of the B color channel and the R color channel, the G color channel and the R color channel are aligned, after which the data is converted by the following specific operations:

$$\begin{cases} Pa(x,y,1,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,3,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,1,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,2,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,3,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,2,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \end{cases} \quad \text{(Equation 18)}$$

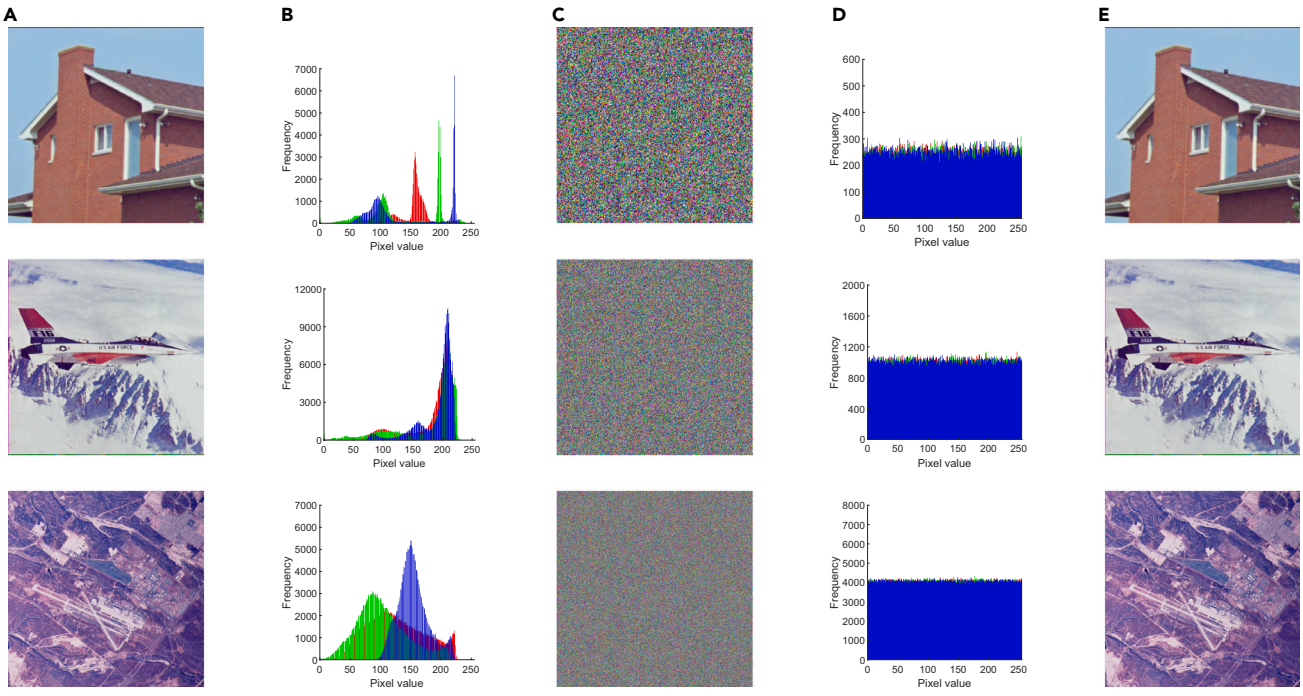


Figure 17. Plaintext and ciphertext images and corresponding histograms

(A) Plain-images.
(B) Histograms of (A).
(C) Encryption results of (A).
(D) Histograms of (C).
(E) Decryption results of (C).

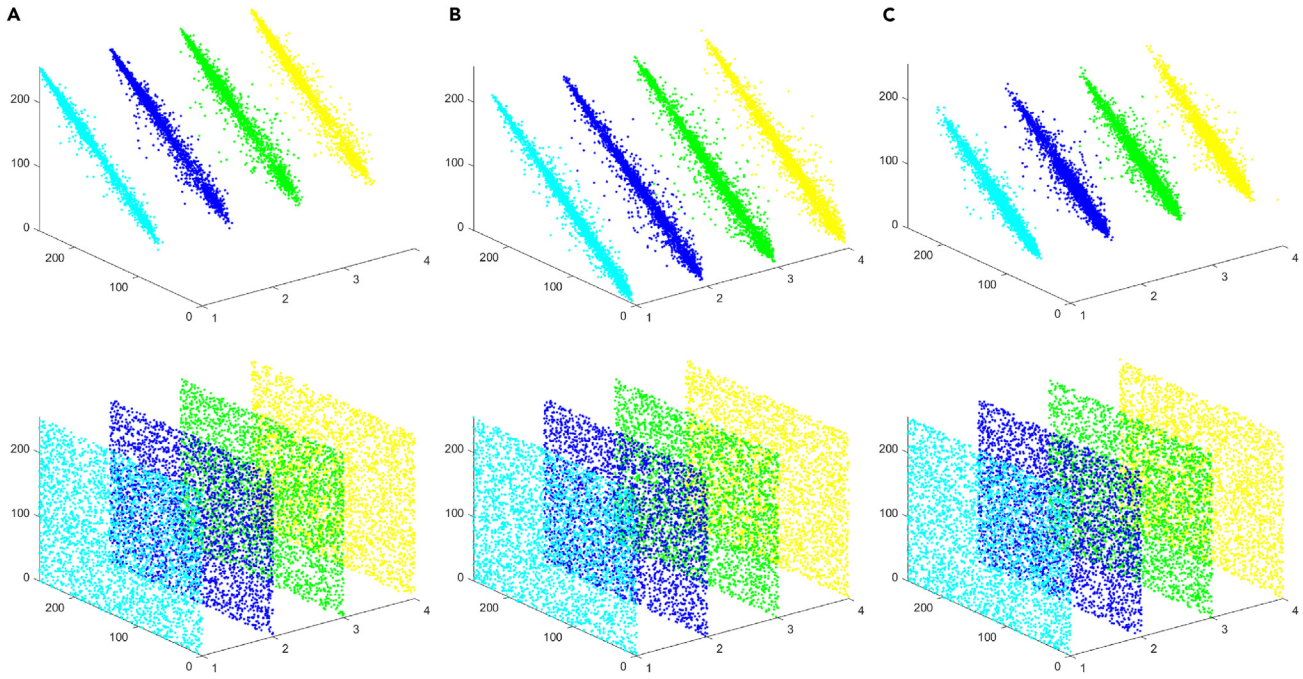


Figure 18. Adjacent pixels' correlation of plaintext image and ciphertext image

(A) R channel.

(B) G channel.

(C) B channel.

- when $S_4' = 5$, the three-dimensional components of color are transformed from *RGB* to *GRB* and then to *GBR*. Following the alignment of the R color channel and the G color channel, the R color channel and the B color channel are aligned, and the data is converted by the following specific operations:

$$\begin{cases} Pa(x,y,1,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \\ Pa(x,y,2,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \\ Pa(x,y,1,i,j) = bitxor(Pa(x,y,1,i,j), Pa(x,y,2,i,j)) \\ Pa(x,y,2,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,3,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \\ Pa(x,y,2,i,j) = bitxor(Pa(x,y,2,i,j), Pa(x,y,3,i,j)) \end{cases} \quad \text{(Equation 19)}$$

Table 1. The comparison results of the correlation coefficients of adjacent pixels

Component	Direction	Original Image	Proposed
R channel	Horizontal	0.9903	−0.0080
	Vertical	0.9793	0.0088
	Diagonal	0.9692	−0.0025
	Anti-diagonal	0.9782	−0.0163
G channel	Horizontal	0.9786	−0.0566
	Vertical	0.9837	0.0032
	Diagonal	0.9725	0.0131
	Anti-diagonal	0.9559	0.0044
B channel	Horizontal	0.9602	−0.0162
	Vertical	0.9292	0.0426
	Diagonal	0.9183	−0.0140
	Anti-diagonal	0.9334	−0.0019

Table 2. Information entropy values of different images

Filename	Description	HI	HC
4.1.01	Female (NTSC test image)	6.8981	7.9991
4.1.02	Couple (NTSC test image)	6.2945	7.9990
4.1.03	Female (from Bell Labs?)	5.9709	7.9991
4.1.04	Female	7.4270	7.9992
4.1.05	House	7.0686	7.9990
4.1.06	Tree	7.5371	7.9989
4.1.07	Jelly beans	6.5835	7.9990
4.1.08	Jelly beans	6.8527	7.9989
4.2.01	Splash	7.2428	7.9998
4.2.03	Mandrill (a.k.a. Baboon)	7.7624	7.9998
4.2.05	Airplane (F-16)	6.6639	7.9998
4.2.06	Sailboat on lake	7.7622	7.9997
4.2.07	Peppers	7.6698	7.9997
2.1.01	San Diego (Miramar NAS)	7.4705	7.9998
2.1.02	San Diego	7.3311	7.9998
2.1.03	San Francisco (Golden Gate)	6.8709	7.9998
2.1.04	Oakland	7.0728	7.9998
2.1.05	San Diego (North Island NAS)	7.5787	7.9998
2.1.06	Woodland Hills, Ca.	7.3983	7.9998
2.1.07	Foster City, Ca.	7.3983	7.9998
2.1.08	San Diego	6.3833	7.9997
2.1.09	San Diego (Point Loma)	6.8026	7.9998
2.1.10	San Diego (Shelter Island)	6.9127	7.9998
2.1.11	Earth from space	7.3511	7.9998
2.1.12	San Diego (Downtown)	6.5674	7.9998
2.2.01	San Diego	7.6133	7.9999
2.2.02	San Diego	6.5786	7.9999
2.2.03	San Diego	6.3301	7.9999
2.2.04	Richmond, Ca.	6.9188	7.9999
2.2.05	San Diego (Miramar NAS)	7.4349	7.9999
2.2.06	San Francisco (Bay Bridge)	6.4133	7.9999
2.2.07	Oakland	6.4492	7.9999
2.2.08	San Diego	7.6470	7.9999
2.2.09	San Francisco	6.8672	7.9999
2.2.10	Richmond and San Rafael	6.7117	7.9999
2.2.11	Stockton	6.7947	7.9999
2.2.12	San Francisco and Oakland	6.9619	7.9999
2.2.13	Stockton	7.3534	7.9999
2.2.14	Shreveport	7.0507	7.9999
2.2.15	San Francisco	6.7776	7.9999
2.2.16	San Francisco	7.1279	7.9999
2.2.17	San Francisco	7.1177	7.9999
2.2.18	Stockton	6.9236	7.9999
2.2.19	Stockton	6.6772	7.9999
2.2.20	Stockton	7.0353	7.9999

(*Continued on next page*)

Table 2. *Continued*

Filename	Description	HI	HC
2.2.21	San Francisco	6.9940	7.9999
2.2.22	San Francisco and Oakland	6.7777	7.9999
2.2.23	San Diego	6.3572	7.9999
2.2.24	Stockton	7.3338	7.9999

- when $S_4' = 0$, the RGB order of the input image remains unprocessed.

Step8: To process the 5D array *Pa* back to a 3D array, the specific operations are as follows:

$$Cp((i-1)\times Px+1:i\times Px,(j-1)\times Px+1:j\times Px,:)=Pa(:,:,:,i,j) \quad \text{(Equation 20)}$$

Finally, the ciphertext image *Cp* is obtained after the chaos-driven three-dimensional partition permutation.

Irregular diffusion

To achieve the diffusion property of the encryption algorithm, we used a diffusion algorithm. The processing order is determined by the generated chaotic sequence, meaning that a pixel can be influenced by any pixel in other color planes. The specific operations are outlined below:

$$C_{ii,j,k}=\begin{cases}\left(C_{pi,j,k}+C_{pM,N,3}+A_{i,j,k}\right) \bmod 256 & \text{if } i=1,j=1,k=1\\ \left(C_{pi,j,k}+C_{iM,N,k-1}+A_{i,j,k}\right) \bmod 256 & \text{if } i=1,j=1,k\neq 1\\ \left(C_{pi,j,k}+C_{iM,j-1,k}+A_{ij,k}\right) \bmod 256 & \text{if } i=1,j\neq 1\\ \left(C_{pi,j,k}+C_{ii-1,j,k}+A_{ij,k}\right) \bmod 256 & \text{if } i\neq 1\end{cases} \quad \text{(Equation 21)}$$

where *Cp* is the input color image and *A* is the chaotic matrix generated by the chaotic sequence S_j.

Sequence bidirectional staggered confusion

Existing confusion operations are typically processed sequentially on a pixel-by-pixel basis, which does not fully utilize computational parallelism. To address this issue, we have implemented a design that utilizes parallel processing. Figure 16 shows the order of the operations and Algorithm 1 introduces the specific operation process.

Overall, the encryption process is hereby completed. And *C* is the final ciphertext image.

RESULTS AND ANALYSIS

Experimental platform

The experiment used a computer running the Windows 11 operating system and equipped with the MATLAB R2023b experimental software. The host is equipped with an 11th Gen Intel Core i7-11800H @ 2.30 GHz eight-core processor and 32 GB of memory. All images used in this article are from the USC-SIPI and Pexels database.

Table 3. Comparison of information entropy between different algorithms

Filename	Proposed	Zhang et al.[41]	Peng et al.[42]	Zhang and Hu[43]	Song et al.[44]
Airplane	7.9998	7.9983	7.9994	7.9992	/
Couple	7.9990	7.9987	/	/	7.9973
House	7.9990	7.9988	7.9978	7.9994	7.9968
Mandrill	7.9998	7.9986	/	7.9992	7.9992
Peppers	7.9997	7.9992	7.9994	7.9989	7.9971
San Diego	7.9998	7.9995	7.9998	/	/
Tree	7.9989	7.9994	/	/	/
Female	7.9992	/	7.9974	/	7.9971
Oakland	7.9999	/	7.9998	/	/
Stockton	7.9999	/	7.9998	/	/

Table 4. MSE, PSNR, and SSIM values of different images

Filename	Size	MSE	PSNR	SSIM
4.1.01	256	36379.0000	2.5223	0.0082
4.1.02	256	46247.0000	1.4800	0.0052
4.1.03	256	19769.0000	5.1709	0.0099
4.1.04	256	25468.0000	4.0709	0.0093
4.1.05	256	25058.0000	4.1413	0.0111
4.1.06	256	29732.0000	3.3986	0.0132
4.1.07	256	27010.0000	3.8155	0.0106
4.1.08	256	26639.0000	3.8757	0.0103
4.2.01	512	33629.0000	2.8637	0.0097
4.2.03	512	25797.0000	4.0151	0.0114
4.2.05	512	31054.0000	3.2097	0.0096
4.2.06	512	30328.0000	3.3124	0.0100
4.2.07	512	30302.0000	3.3160	0.0096
2.1.01	512	22328.0000	4.6423	0.0109
2.1.02	512	25417.0000	4.0796	0.0100
2.1.03	512	28461.0000	3.5883	0.0058
2.1.04	512	22496.0000	4.6098	0.0084
2.1.05	512	24661.0000	4.2107	0.0093
2.1.06	512	23237.0000	4.4689	0.0109
2.1.07	512	23237.0000	4.4689	0.0109
2.1.08	512	27995.0000	3.6600	0.0099
2.1.09	512	26393.0000	3.9160	0.0108
2.1.10	512	26627.0000	3.8776	0.0109
2.1.11	512	22605.0000	4.5888	0.0105
2.1.12	512	27889.0000	3.6765	0.0108
2.2.01	1024	27657.0000	3.7127	0.0080
2.2.02	1024	26238.0000	3.9416	0.0106
2.2.03	1024	28353.0000	3.6048	0.0104
2.2.04	1024	27052.0000	3.8088	0.0109
2.2.05	1024	22284.0000	4.6508	0.0106
2.2.06	1024	25786.0000	4.0170	0.0112
2.2.07	1024	27591.0000	3.7231	0.0107
2.2.08	1024	27254.0000	3.7766	0.0094
2.2.09	1024	26243.0000	3.9406	0.0112
2.2.10	1024	26277.0000	3.9351	0.0105
2.2.11	1024	23685.0000	4.3861	0.0106
2.2.12	1024	22190.0000	4.6693	0.0108
2.2.13	1024	23396.0000	4.4394	0.0098
2.2.14	1024	21637.0000	4.7788	0.0106
2.2.15	1024	23083.0000	4.4978	0.0101
2.2.16	1024	21364.0000	4.8339	0.0110
2.2.17	1024	22622.0000	4.5855	0.0108
2.2.18	1024	23532.0000	4.4143	0.0108
2.2.19	1024	24447.0000	4.2486	0.0108

(*Continued on next page*)

Table 4. *Continued*

Filename	Size	MSE	PSNR	SSIM
2.2.20	1024	22628.0000	4.5844	0.0106
2.2.21	1024	26925.0000	3.8293	0.0099
2.2.22	1024	26607.0000	3.8808	0.0101
2.2.23	1024	26033.0000	3.9756	0.0109
2.2.24	1024	23049.0000	4.5043	0.0088

Statistical analysis

Histogram analysis

A sophisticated image encryption algorithm cleverly hides important information by manipulating the image distribution to simulate noise. After our processing, the selected image, as shown in Figure 17, has a histogram distribution that resembles noise after encryption. This effectively eliminates the potential for attackers to extract valuable information.

Correlation analysis

Disrupting the correlation between pixels is a crucial task for an image encryption algorithm. It effectively thwarts attackers' decryption attempts that rely on pixel correlations. This article uses the 'Lena' image as an example to elucidate this process. We randomly selected 3000 pairs of adjacent pixels from both the plaintext and ciphertext. We then calculated the correlation coefficients for these adjacent pixels in horizontal, vertical, diagonal, and anti-diagonal directions. The corresponding scatterplots are depicted in Figure 18. The outcomes of the correlation analysis are presented in Table 1. The correlation coefficient between adjacent pixels can be represented as:

Table 5. NPCR test values for images of size 256

Filename	Description	Size	Channel	NPCR
4.1.01	Female (NTSC test image)	256	Red	99.5636
4.1.01	Female (NTSC test image)	256	Green	99.6048
4.1.01	Female (NTSC test image)	256	Blue	99.6353
4.1.02	Couple (NTSC test image)	256	Red	99.6002
4.1.02	Couple (NTSC test image)	256	Green	99.6201
4.1.02	Couple (NTSC test image)	256	Blue	99.6002
4.1.03	Female (from Bell Labs?)	256	Red	99.6017
4.1.03	Female (from Bell Labs?)	256	Green	99.5865
4.1.03	Female (from Bell Labs?)	256	Blue	99.5956
4.1.04	Female	256	Red	99.5743
4.1.04	Female	256	Green	99.6017
4.1.04	Female	256	Blue	99.5987
4.1.05	House	256	Red	99.6490
4.1.05	House	256	Green	99.6017
4.1.05	House	256	Blue	99.6277
4.1.06	Tree	256	Red	99.5621
4.1.06	Tree	256	Green	99.6704
4.1.06	Tree	256	Blue	99.6338
4.1.07	Jelly beans	256	Red	99.5895
4.1.07	Jelly beans	256	Green	99.5911
4.1.07	Jelly beans	256	Blue	99.6231
4.1.08	Jelly beans	256	Red	99.5789
4.1.08	Jelly beans	256	Green	99.6216
4.1.08	Jelly beans	256	Blue	99.5712

Table 6. NPCR test values for images of size 512

Filename	Description	Size	Channel	NPCR
4.2.01	Splash	512	Red	99.6037
4.2.01	Splash	512	Green	99.6105
4.2.01	Splash	512	Blue	99.5972
4.2.03	Mandrill (a.k.a. Baboon)	512	Red	99.5911
4.2.03	Mandrill (a.k.a. Baboon)	512	Green	99.5995
4.2.03	Mandrill (a.k.a. Baboon)	512	Blue	99.6181
4.2.05	Airplane (F-16)	512	Red	99.6002
4.2.05	Airplane (F-16)	512	Green	99.6269
4.2.05	Airplane (F-16)	512	Blue	99.6098
4.2.06	Sailboat on lake	512	Red	99.5998
4.2.06	Sailboat on lake	512	Green	99.5979
4.2.06	Sailboat on lake	512	Blue	99.6063
4.2.07	Peppers	512	Red	99.6071
4.2.07	Peppers	512	Green	99.6162
4.2.07	Peppers	512	Blue	99.6216
2.1.01	San Diego (Miramar NAS)	512	Red	99.6387
2.1.01	San Diego (Miramar NAS)	512	Green	99.6002
2.1.01	San Diego (Miramar NAS)	512	Blue	99.6269
2.1.02	San Diego	512	Red	99.6101
2.1.02	San Diego	512	Green	99.5899
2.1.02	San Diego	512	Blue	99.6243
2.1.03	San Francisco (Golden Gate)	512	Red	99.6101
2.1.03	San Francisco (Golden Gate)	512	Green	99.6014
2.1.03	San Francisco (Golden Gate)	512	Blue	99.6193
2.1.04	Oakland	512	Red	99.6239
2.1.04	Oakland	512	Green	99.6086
2.1.04	Oakland	512	Blue	99.6105
2.1.05	San Diego (North Island NAS)	512	Red	99.6056
2.1.05	San Diego (North Island NAS)	512	Green	99.6014
2.1.05	San Diego (North Island NAS)	512	Blue	99.6078
2.1.06	Woodland Hills, Ca.	512	Red	99.6155
2.1.06	Woodland Hills, Ca.	512	Green	99.6132
2.1.06	Woodland Hills, Ca.	512	Blue	99.6311
2.1.07	Foster City, Ca.	512	Red	99.6155
2.1.07	Foster City, Ca.	512	Green	99.6132
2.1.07	Foster City, Ca.	512	Blue	99.6311
2.1.08	San Diego	512	Red	99.6101
2.1.08	San Diego	512	Green	99.6136
2.1.08	San Diego	512	Blue	99.6128
2.1.09	San Diego (Point Loma)	512	Red	99.6185
2.1.09	San Diego (Point Loma)	512	Green	99.6273
2.1.09	San Diego (Point Loma)	512	Blue	99.6174
2.1.10	San Diego (Shelter Island)	512	Red	99.5983
2.1.10	San Diego (Shelter Island)	512	Green	99.5834
2.1.10	San Diego (Shelter Island)	512	Blue	99.6048

(*Continued on next page*)

Table 6. *Continued*

Filename	Description	Size	Channel	NPCR
2.1.11	Earth from space	512	Red	99.5914
2.1.11	Earth from space	512	Green	99.6136
2.1.11	Earth from space	512	Blue	99.6078
2.1.12	San Diego (Downtown)	512	Red	99.6162
2.1.12	San Diego (Downtown)	512	Green	99.6105
2.1.12	San Diego (Downtown)	512	Blue	99.6029

$$\begin{cases} r_{xy} = \dfrac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \\ \mathrm{cov}(x,y) = \dfrac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \\ D(x) = \dfrac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \\ E(x) = \dfrac{1}{N}\sum_{i=1}^{N}x_i \end{cases} \quad \text{(Equation 22)}$$

where x_i and y_i constitute the i-th pair of neighboring pixels, N is the total number of neighboring pixels, $cov(x,y)$ is the covariance between pixel values x and y, $D(x)$ and $D(y)$ are the pixel value x and pixel value y mean-square error, $E(x)$ and $E(y)$ are the expected values of pixel value x and pixel value y, respectively. And r_{xy} is the correlation coefficient of pixel values x and y.

The experiment shows that our encryption algorithm effectively reduces the correlation between pixels, resulting in almost imperceptible correlation in the ciphertext.

Information entropy analysis

Information entropy is an important indicator of the uncertainty of image information. When the uncertainty within an image increases, its information entropy also increases, which enhances confidentiality. Conversely, a decrease in information entropy indicates a reduction in the image's uncertainty, leading to a decrease in confidentiality. The equation for calculating information entropy is as follows:

$$H(x) = -\sum_{i=1}^{L} P(x_i)log_2P(x_i) \quad \text{(Equation 23)}$$

where L is the total number of symbols $x_i \in x$ and $P(x_i)$ denotes the probability of the symbols.

Table 2 presents the information entropy results of this study, where HI represents the plaintext image information entropy, and HC represents the ciphertext image information entropy. The information entropy of the ciphertext approaches the ideal value of 8 after encryption, indicating excellent algorithm performance in terms of confidentiality. Table 3 compares the data from this article with other approaches.

Image quality analysis

Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are frequently used to evaluate the quality of encryption in image processing. PSNR includes mean square error (MSE) as a component, which is defined as:

$$\begin{cases} \mathrm{MSE} = \dfrac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}(X(i,j) - Y(i,j))^2 \\ \mathrm{PSNR} = 10 \times \log_{10}\left(\dfrac{Q^2}{MSE}\right) \end{cases} \quad \text{(Equation 24)}$$

where MSE represents the mean square error between the plaintext image X and the ciphertext image Y, with H and W representing the height and width of the image, respectively. Q denotes the pixel level of the image. SSIM is a metric that quantifies the similarity between two images, defined as:

$$\mathrm{SSIM}(X,Y) = \frac{\left(2\mu_X\mu_Y+(0.01L)^2\right)\left(2\mu_{XY}+(0.03L)^2\right)}{\left(\mu_X^2+\mu_Y^2+(0.01L)^2\right)\left(\mu_X^2+\mu_Y^2+(0.03L)^2\right)} \quad \text{(Equation 25)}$$

Table 7. NPCR test values for images of size 1024

Filename	Description	Size	Channel	NPCR
2.2.01	San Diego	1024	Red	99.6017
2.2.01	San Diego	1024	Green	99.6174
2.2.01	San Diego	1024	Blue	99.6077
2.2.02	San Diego	1024	Red	99.6071
2.2.02	San Diego	1024	Green	99.6172
2.2.02	San Diego	1024	Blue	99.6122
2.2.03	San Diego	1024	Red	99.6010
2.2.03	San Diego	1024	Green	99.6131
2.2.03	San Diego	1024	Blue	99.6095
2.2.04	Richmond, Ca.	1024	Red	99.6145
2.2.04	Richmond, Ca.	1024	Green	99.6162
2.2.04	Richmond, Ca.	1024	Blue	99.6024
2.2.05	San Diego (Miramar NAS)	1024	Red	99.5968
2.2.05	San Diego (Miramar NAS)	1024	Green	99.6024
2.2.05	San Diego (Miramar NAS)	1024	Blue	99.6305
2.2.06	San Francisco (Bay Bridge)	1024	Red	99.6078
2.2.06	San Francisco (Bay Bridge)	1024	Green	99.6099
2.2.06	San Francisco (Bay Bridge)	1024	Blue	99.6036
2.2.07	Oakland	1024	Red	99.6194
2.2.07	Oakland	1024	Green	99.6093
2.2.07	Oakland	1024	Blue	99.6163
2.2.08	San Diego	1024	Red	99.6169
2.2.08	San Diego	1024	Green	99.6091
2.2.08	San Diego	1024	Blue	99.6153
2.2.09	San Francisco	1024	Red	99.6119
2.2.09	San Francisco	1024	Green	99.6153
2.2.09	San Francisco	1024	Blue	99.6025
2.2.10	Richmond and San Rafael	1024	Red	99.6116
2.2.10	Richmond and San Rafael	1024	Green	99.6074
2.2.10	Richmond and San Rafael	1024	Blue	99.6157
2.2.11	Stockton	1024	Red	99.6050
2.2.11	Stockton	1024	Green	99.6008
2.2.11	Stockton	1024	Blue	99.6136
2.2.12	San Francisco and Oakland	1024	Red	99.6028
2.2.12	San Francisco and Oakland	1024	Green	99.6106
2.2.12	San Francisco and Oakland	1024	Blue	99.6032
2.2.13	Stockton	1024	Red	99.6188
2.2.13	Stockton	1024	Green	99.6126
2.2.13	Stockton	1024	Blue	99.6035
2.2.14	Shreveport	1024	Red	99.6032
2.2.14	Shreveport	1024	Green	99.6158
2.2.14	Shreveport	1024	Blue	99.6060
2.2.15	San Francisco	1024	Red	99.6090
2.2.15	San Francisco	1024	Green	99.6082
2.2.15	San Francisco	1024	Blue	99.6139

(*Continued on next page*)

Table 7. *Continued*

Filename	Description	Size	Channel	NPCR
2.2.16	San Francisco	1024	Red	99.6090
2.2.16	San Francisco	1024	Green	99.6163
2.2.16	San Francisco	1024	Blue	99.6175
2.2.17	San Francisco	1024	Red	99.6041
2.2.17	San Francisco	1024	Green	99.6109
2.2.17	San Francisco	1024	Blue	99.6103
2.2.18	Stockton	1024	Red	99.5985
2.2.18	Stockton	1024	Green	99.6150
2.2.18	Stockton	1024	Blue	99.6093
2.2.19	Stockton	1024	Red	99.6112
2.2.19	Stockton	1024	Green	99.6027
2.2.19	Stockton	1024	Blue	99.6062
2.2.20	Stockton	1024	Red	99.6054
2.2.20	Stockton	1024	Green	99.6195
2.2.20	Stockton	1024	Blue	99.6143
2.2.21	San Francisco	1024	Red	99.6137
2.2.21	San Francisco	1024	Green	99.6019
2.2.21	San Francisco	1024	Blue	99.6056
2.2.22	San Francisco and Oakland	1024	Red	99.6110
2.2.22	San Francisco and Oakland	1024	Green	99.6089
2.2.22	San Francisco and Oakland	1024	Blue	99.6134
2.2.23	San Diego	1024	Red	99.6141
2.2.23	San Diego	1024	Green	99.6037
2.2.23	San Diego	1024	Blue	99.6011
2.2.24	Stockton	1024	Red	99.6273
2.2.24	Stockton	1024	Green	99.6122
2.2.24	Stockton	1024	Blue	99.6112

where the mean values of the images X and Y are denoted by μ_X and μ_Y, respectively, while their respective standard deviations are represented as well. The dynamic range of pixel values is denoted as L.

Equations 24 and 25 are used to calculate the values of MSE, PSNR, and SSIM. Additionally, to ensure generality, several images were selected to test the encryption module. The encrypted images should have a PSNR of less than 10 dB and an SSIM value close to 0. Table 4 details the results of the three tests. The experimental results demonstrate the excellent encryption performance of our algorithm.

Table 8. Comparison of NPCR values between different algorithms

Filename	Proposed	Zhang et al.[41]	Peng et al.[42]	Zhang and Hu[43]	Song et al.[44]
Airplane	99.6098	99.6283	99.6330	99.6092	/
Couple	99.6002	99.5845	/	/	99.6130
House	99.6017	99.6296	99.6399	99.6128	99.6110
Mandrill	99.6181	99.6296	/	99.6131	99.6110
Peppers	99.6071	99.6236	99.6174	99.6071	/
San Diego	99.6101	99.6291	99.6172	/	/
Tree	99.6338	99.6074	/	/	/
Female	99.6017	/	99.5880	/	/
Oakland	99.6093	/	99.6147	/	/
Stockton	99.6050	/	99.6066	/	/

Table 9. Key sensitivity test

Description	Size	Channel	$a+10^{-15}$ NPCR	$a+10^{-15}$ UACI	$b+10^{-15}$ NPCR	$b+10^{-15}$ UACI	$c+10^{-15}$ NPCR	$c+10^{-15}$ UACI	$d+10^{-15}$ NPCR	$d+10^{-15}$ UACI
House	256	Red	99.6307	33.5170	99.5743	33.5130	99.5499	33.4093	99.6155	33.3815
House	256	Green	99.5865	33.3260	99.6567	33.3445	99.5758	33.3546	99.6262	33.3243
House	256	Blue	99.6017	33.5700	99.5926	33.3201	99.6185	33.5894	99.6063	33.4627
Airplane (F-16)	512	Red	99.6174	33.4355	99.5720	33.3455	99.6021	33.3960	99.6120	33.4653
Airplane (F-16)	512	Green	99.6025	33.4237	99.6044	33.3569	99.6143	33.3931	99.6101	33.4920
Airplane (F-16)	512	Blue	99.5892	33.4288	99.6101	33.4242	99.5991	33.4436	99.6109	33.4780
San Diego	1024	Red	99.6053	33.4061	99.6053	33.4301	99.6065	33.4460	99.6078	33.4842
San Diego	1024	Green	99.5965	33.4564	99.6023	33.4271	99.5928	33.4474	99.6162	33.4703
San Diego	1024	Blue	99.6129	33.4188	99.6026	33.4285	99.6117	33.4385	99.6124	33.4538

Differential statistical analysis

The number-of-pixels change rate (NPCR) serves as a widely used metric for comparing the similarity of two images, defined as the count of changed pixels between them. It is frequently employed to evaluate the performance of encryption algorithms against differential attacks, which are among the most effective methods for block cipher attacks. In a typical scenario, the attacker seeks to extract the key by analyzing the impact of specific plaintext differences on corresponding ciphertext differences, thereby attempting to compromise the algorithm's security. To assess the proposed algorithm's resilience against differential attacks, we perform numerical calculations and compare NPCR using Equation 26. Due to the large amount of data, the images are grouped by size, with each group size being 256, 512 and 1024, corresponding to Tables 5, 6, and 7, respectively. Meanwhile, Table 8 compares the data from this article with other approaches. The NPCR equation is expressed as follows:

$$NPCR = \frac{1}{M \times N} \times \sum_{i=1}^{M} \sum_{i=1}^{N} D(i,j) \times 100\% \quad \text{(Equation 26)}$$

where $M \times N$ is the size of the image. D can be defined by the following equation:

$$D(i,j) = \begin{cases} 0, v_1(i,j) = v_2(i,j) \\ 1, v_0(i,j) \neq v_2(i,j) \end{cases} \quad \text{(Equation 27)}$$

Analysis of sensitivity to the key

Key sensitivity is a crucial aspect of encryption technology, referring to the significant differences observed in ciphertexts when encrypting the same image with slightly varying keys. In this section, we use the original key (key) and a key with a small perturbation ($key + 10^{-15}$) to encrypt the same plaintext. The generated ciphertexts are compared, and the NPCR and UACI are calculated using Equations 26 and 28, respectively. Table 9 presents the experimental results. The results indicate significant differences between the two ciphertext images obtained with the perturbed key. The NPCR and UACI values are close to the ideal values of 99.6094 and 33.4635, respectively, indicating the encryption scheme's high sensitivity to the key. The UACI equation is:

$$UACI = \frac{1}{M \times N} \times \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|v_1(i,j) - v_2(i,j)|}{255} \times 100\% \quad \text{(Equation 28)}$$

where $v_1(i,j)$, $v_2(i,j)$ are the ciphertext images before and after changing one pixel of the plaintext image, respectively.

Key space

The key space represents the range of the encryption key, and having a sufficiently large key space is crucial for effective protection against exhaustive attacks. The algorithm specifically chooses eight key parameters with an accuracy of 10^{-15}. Therefore, the key space of the algorithm can be estimated to be about $10^{15\times8} = 10^{120}$, which exceeds the theoretical requirement of 2^{100}. At the same time, we compared the key spaces, as shown in Table 10.

Conclusion

This article presents a secure communication enhancement scheme based on face detection and chaos-driven three-dimensional partition permutation. The scheme provides an effective approach for dealing with privacy breaches. The face detection technology of edge recognition is used to identify facial information. Then, the hash feature values of the plaintext image are extracted as the key to the chaotic system to generate the required chaotic sequence for encryption. The ciphertext image is obtained through subsequent partition permutation, irregular diffusion, and bidirectional parallel confusion. In the experimental section, we conducted a thorough security assessment of the proposed

Table 10. Key space comparison

Proposed	Gan[45]	Zhang et al[46]	Wen et al[47]	Rehman et al[48]	Ponnaian and Chandranbabu[49]
10^{120}	10^{79}	2^{299}	2^{287}	2^{200}	10^{56}

algorithm and achieved good results through the analysis of various experimental data. The proposed algorithm demonstrates superior encryption performance and resistance to typical attacks, effectively safeguarding image privacy while meeting modern communication needs. Future plans include expanding our work to maintain high encryption security.

RESOURCE AVAILABILITY

Lead contact

Further information for resources and materials should be directed to and will be fulfilled by the lead contact, Dr. H.W. (wenheping@uestc.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All experimental data is clearly explained in this article.
- This article does not report original code.
- Any additional information required to reanalyze the data reported in this article is available from the lead contact upon request.

ACKNOWLEDGMENTS

This work was supported in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011717, Special Projects for Key Fields of the Education Department of Guangdong Province under Grant 2023ZDZX1041, the Key Laboratory of Guangdong Higher Education Institutes under Grant 2023KSYS011, and Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation ("Climbing Program" Special Funds) under Grant pdjh2024a428 and pdjh2023b0600.

AUTHOR CONTRIBUTIONS

Z.X.: Conceptualization, Methodology, Software, Visualization, Data Curation, Writing - Original Draft, Writing - Review and Editing. Y.L.: Conceptualization, Methodology, Project Administration, Resources, Writing - Original Draft, Writing - Review and Editing. T.L.: Visualization, Formal Analysis. H.W.: Funding acquisition.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - Adjacent pixel correlation analysis
  - Differential statistical analysis
  - Information entropy analysis
  - Image quality analysis

Received: April 7, 2024
Revised: July 4, 2024
Accepted: August 15, 2024
Published: August 20, 2024

REFERENCES

1. Gao, S., Iu, H.H.C., Mou, J., Erkan, U., Liu, J., Wu, R., and Tang, X. (2024). Temporal action segmentation for video encryption. Chaos, Solit. Fractals *183*, 114958. https://doi.org/10.1016/j.chaos.2024.114958.
2. Gao, S., Liu, J., Ho-Ching Iu, H., Erkan, U., Zhou, S., Wu, R., and Tang, X. (2024). Development of a video encryption algorithm for critical areas using 2d extended schaffer function map and neural networks. Appl. Math. Model. *134*, 520–537. https://doi.org/10.1016/j.apm.2024.06.016.
3. Lin, Y., Xie, Z., Chen, T., Cheng, X., and Wen, H. (2024). Image privacy protection scheme based on high-quality reconstruction dct compression and nonlinear dynamics. Expert Syst. Appl. *257*, 124891. https://doi.org/10.1016/j.eswa.2024.124891.
4. Neamah, A.A. (2023). An image encryption scheme based on a seven-dimensional hyperchaotic system and pascal's matrix. J. King Saud Univ. Comp. Inform. Sci. *35*,

238–248. https://doi.org/10.1016/j.jksuci.2023.02.014.

5. Zhang, Y., Zhao, R., Xiao, X., Lan, R., Liu, Z., and Zhang, X. (2022). Hf-tpe: High-fidelity thumbnail- preserving encryption. IEEE Trans. Circuits Syst. Video Technol. *32*, 947–961. https://doi.org/10.1109/TCSVT.2021.3070348.
6. Zhang, Y., Zhou, W., Zhao, R., Zhang, X., and Cao, X. (2023). F-tpe: Flexible thumbnail-preserving encryption based on multi-pixel sum-preserving encryption. IEEE Trans. Multimedia *25*, 5877–5891. https://doi.org/10.1109/TMM.2022.3200310.
7. Feng, W., Wang, Q., Liu, H., Ren, Y., Zhang, J., Zhang, S., Qian, K., and Wen, H. (2023). Exploiting newly designed fractional-order 3d lorenz chaotic system and 2d discrete polynomial hyper-chaotic map for high-performance multi-image encryption. Fractal Fract. *7*, 887. https://doi.org/10.3390/fractalfract7120887.
8. Feng, W., Zhao, X., Zhang, J., Qin, Z., Zhang, J., and He, Y. (2022). Image encryption algorithm based on plane-level image filtering and discrete logarithmic transform. Mathematics *10*, 2751. https://doi.org/10.3390/math10152751.
9. Winarno, E., Nugroho, K., Adi, P.W., and Setiadi, D.R.I.M. (2023). Integrated dual hyperchaotic and josephus traversing based 3d confusion-diffusion pattern for image encryption. J. King Saud Univ. Comp. Inform. Sci. *35*, 101790. https://doi.org/10.1016/j.jksuci.2023.101790.
10. Li, C., Tan, K., Feng, B., and Lu, J. (2022). The graph structure of the generalized discrete arnold's cat map. IEEE Trans. Comput. *71*, 364–377. https://doi.org/10.1109/TC.2021.3051387.
11. Teng, L., Wang, X., and Xian, Y. (2022). Image encryption algorithm based on a 2d-clss hyperchaotic map using simultaneous permutation and diffusion. Inf. Sci. *605*, 71–85. https://doi.org/10.1016/j.ins.2022.05.032.
12. Wen, H., Lin, Y., Xie, Z., and Liu, T. (2023). Chaos-based block permutation and dynamic sequence multiplexing for video encryption. Sci. Rep. *13*, 14721. https://doi.org/10.1038/s41598-023-41082-9.
13. Liu, H., Teng, L., Zhang, Y., Si, R., and Liu, P. (2024). Mutil-medical image encryption by a new spatiotemporal chaos model and dna new computing for information security. Expert Syst. Appl. *235*, 121090. https://doi.org/10.1016/j.eswa.2023.121090.
14. Liu, L., Zhang, Q., and Wei, X. (2012). A rgb image encryption algorithm based on dna encoding and chaos map. Comput. Electr. Eng. *38*, 1240–1248. Special issue on Recent Advances in Security and Privacy in Distributed Communications and Image processing. https://doi.org/10.1016/j.compeleceng.2012.02.007.
15. Wang, X., and Li, Y. (2021). Chaotic image encryption algorithm based on hybrid multi-objective particle swarm optimization and dna sequence. Opt Laser. Eng. *137*, 106393. https://doi.org/10.1016/j.optlaseng.2020.106393.
16. Begum, M., Ferdush, J., and Uddin, M.S. (2022). A hybrid robust watermarking system based on discrete cosine transform, discrete wavelet transform, and singular value decomposition. J. King Saud Univ.-Comp. Inform. Sci. *34*, 5856–5867. https://doi.org/10.1016/j.jksuci.2021.07.012.
17. Rehman, M.U., Shafique, A., Khan, K.H., and Hazzazi, M.M. (2023). Efficient and secure image encryption using key substitution process with discrete wavelet transform. J. King Saud Univ.-Comp. Inform. Sci. *35*, 101613. https://doi.org/10.1016/j.jksuci.2023.101613.
18. Xie, H., Lu, J., Han, J., Zhang, Y., Xiong, F., and Zhao, Z. (2023). Fourier coded aperture transform hyperspectral imaging system. Opt Laser. Eng. *163*, 107443. https://doi.org/10.1016/j.optlaseng.2022.107443.
19. Luo, Y., Liang, X., Zhang, C., Zeng, W., and Qiu, K. (2024). Redundancy-free key distribution using multiple phase offset for secure data center. J. Lightwave Technol. *42*, 523–531. https://doi.org/10.1109/JLT.2023.3320037.
20. Teng, L., Wang, X., Yang, F., and Xian, Y. (2021). Color image encryption based on cross 2d hyperchaotic map using combined cycle shift scrambling and selecting diffusion. Nonlinear Dyn. *105*, 1859–1876. https://doi.org/10.1007/s11071-021-06663-1.
21. Zeng, W., Zhang, C., Liang, X., Luo, Y., Wang, X., and Qiu, K. (2024). Chaotic phase noise-like encryption based on geometric shaping for coherent data center interconnections. Opt Express *32*, 1595–1608. https://doi.org/10.1364/OE.506738.
22. Kocak, O., Erkan, U., Toktas, A., and Gao, S. (2024). Pso-based image encryption scheme using modular integrated logistic exponential map. Expert Syst. Appl. *237*, 121452. https://doi.org/10.1016/j.eswa.2023.121452.
23. Liang, X., Zhang, C., Luo, Y., Wang, X., and Qiu, K. (2023). Secure encryption and key management for ofdm-pon based on chaotic hilbert motion. J. Lightwave Technol. *41*, 1619–1625. https://doi.org/10.1109/JLT.2022.3226768.
24. Wen, H., Xie, Z., Wu, Z., Lin, Y., and Feng, W. (2024). Exploring the future application of uavs: Face image privacy protection scheme based on chaos and dna cryptography. J. King Saud Univ. Comp. Inform. Sci. *36*, 101871. https://doi.org/10.1016/j.jksuci.2023.101871.
25. Alawida, M. (2023). A novel chaos-based permutation for image encryption. J. King Saud Univ. Comp. Inform. Sci. *35*, 101595. https://doi.org/10.1016/j.jksuci.2023.101595.
26. Banerjee, M., Ghosh, S., Manfredi, P., and d'Onofrio, A. (2023). Spatio-temporal chaos and clustering induced by nonlocal information and vaccine hesitancy in the sir epidemic model. Chaos, Solit. Fractals *170*, 113339. https://doi.org/10.1016/j.chaos.2023.113339.
27. Melman, A., and Evsutin, O. (2023). Comparative study of metaheuristic optimization algorithms for image steganography based on discrete fourier transform domain. Appl. Soft Comput. *132*, 109847. https://doi.org/10.1016/j.asoc.2022.109847.
28. Erkan, U., Toktas, A., and Lai, Q. (2023). 2d hyperchaotic system based on schaffer function for image encryption. Expert Syst. Appl. *213*, 119076. https://doi.org/10.1016/j.eswa.2022.119076.
29. Erkan, U., Toktas, A., Memiş, S., Lai, Q., and Hu, G. (2023). An image encryption method based on multi-space confusion using hyperchaotic 2d vincent map derived from optimization benchmark function. Nonlinear Dyn. *111*, 20377–20405. https://doi.org/10.1007/s11071-023-08859-z.
30. Toktas, A., Erkan, U., Gao, S., and Pak, C. (2024). A robust bit-level image encryption based on bessel map. Appl. Math. Comput. *462*, 128340. https://doi.org/10.1016/j.amc.2023.128340.
31. Gao, S., Iu, H.H.C., Wang, M., Jiang, D., El-Latif, A.A.A., Wu, R., and Tang, X. (2024). Design, hardware implementation, and application in video encryption of the 2-d memristive cubic map. IEEE Internet Things J. *11*, 21807–21815. https://doi.org/10.1109/JIOT.2024.3376572.
32. Wen, H., Lin, Y., Yang, L., and Chen, R. (2024). Cryptanalysis of an image encryption scheme using variant hill cipher and chaos. Expert Syst. Appl. *250*, 123748. https://doi.org/10.1016/j.eswa.2024.123748.
33. Gao, S., Wu, R., Wang, X., Liu, J., Li, Q., and Tang, X. (2023). Efr-cstp: Encryption for face recognition based on the chaos and semi-tensor product theory. Inf. Sci. *621*, 766–781. https://doi.org/10.1016/j.ins.2022.11.121.
34. Du, L., Teng, L., Liu, H., and Lu, H. (2024). Multiple face images encryption based on a new non-adjacent dynamic coupled mapping lattice. Expert Syst. Appl. *238*, 121728. https://doi.org/10.1016/j.eswa.2023.121728.
35. Li, X., Chen, Z., and Gao, J. (2024). Ciphertext face recognition system based on secure inner product protocol. J. Inf. Secur. Appl. *80*, 103681. https://doi.org/10.1016/j.jisa.2023.103681.
36. Wen, H., and Lin, Y. (2024). 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.
37. Wen, H., and Lin, Y. (2023). Cryptanalyzing an image cipher using multiple chaos and DNA operations. J. King Saud Univ. Comp. Inform. Sci. *35*, 101612. https://doi.org/10.1016/j.jksuci.2023.101612.
38. Wen, H., Lin, Y., and Feng, Z. (2024). Cryptanalyzing a bit-level image encryption algorithm based on chaotic maps. Eng. Sci. Technol. Int. J. *51*, 101634. https://doi.org/10.1016/j.jestch.2024.101634.
39. Li, C., and Yu, S. (2012). A new hyperchaotic system and its adaptive tracking control. Acta Phys. Sin. *61*, 22–28. https://doi.org/10.7498/aps.61.040504.
40. Hua, Z., Jin, F., Xu, B., and Huang, H. (2018). 2d logistic-sine-coupling map for image encryption. Signal Process. *149*, 148–161. https://doi.org/10.1016/j.sigpro.2018.03.010.
41. Zhang, H., Hu, H., and Ding, W. (2024). Vsdhs-ciea: Color image encryption algorithm based on novel variable-structure discrete hyperchaotic system and cross-plane confusion strategy. Inf. Sci. *665*, 120332. https://doi.org/10.1016/j.ins.2024.120332.
42. Peng, Y., Lan, Z., Sun, K., and Xu, W. (2023). A simple color image encryption algorithm based on a discrete memristive hyperchaotic map and time-controllable operation. Opt Laser. Technol. *165*, 109543. https://doi.org/10.1016/j.optlastec.2023.109543.
43. Zhang, H., and Hu, H. (2024). An image encryption algorithm based on a compound-coupled chaotic system. Digit. Signal Process. *146*, 104367. https://doi.org/10.1016/j.dsp.2023.104367.
44. Song, W., Fu, C., Zheng, Y., Zhang, Y., Chen, J., and Wang, P. (2024). Batch image

encryption using cross image permutation and diffusion. J. Inf. Secur. Appl. *80*, 103686. https://doi.org/10.1016/j.jisa.2023.103686.

45. Gan, Z., Chai, X., Zhang, J., Zhang, Y., and Chen, Y. (2020). An effective image compression–encryption scheme based on compressive sensing (cs) and game of life (gol). Neural Comput. Appl. *32*, 14113–14141. https://doi.org/10.1007/s00521-020-04808-8.
46. Zhang, W., Xu, J., and Zhao, B. (2023). Dna image encryption algorithm based on serrated spiral scrambling and cross bit plane. J. King Saud Univ. Comp. Inform. Sci. *35*, 101858. https://doi.org/10.1016/j.jksuci.2023.101858.
47. Wen, H., Lin, Y., Kang, S., Zhang, X., and Zou, K. (2024). Secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion. iScience *27*, 108610. https://doi.org/10.1016/j.isci.2023.108610.
48. Rehman, M.U. (2024). Quantum-enhanced chaotic image encryption: Strengthening digital data security with 1-d sine-based chaotic maps and quantum coding. J. King Saud Univ.-Comp. Inform. Sci. *36*, 101980. https://doi.org/10.1016/j.jksuci.2024.101980.
49. Ponnaian, D., and Chandranbabu, K. (2017). Crypt analysis of an image compression–encryption algorithm and a modified scheme using compressive sensing. Optik *147*, 263–276. https://doi.org/10.1016/j.ijleo.2017.07.063.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
MATLAB R2023b	MathWorks.Inc	https://ww2.mathworks.cn/products/matlab.html
The USC-SIPI Image Database	University of Southern California	https://sipi.usc.edu/database/database.php
The Pexels Image Database	Canva Germany GmbH	https://www.pexels.com

METHOD DETAILS

The experiment used a computer running the Windows 11 operating system and equipped with the MATLAB R2023b experimental software. The host is equipped with an 11th Gen Intel Core i7-11800H @ 2.30 GHz eight-core processor and 32 GB of memory. All images used in this paper are from the USC-SIPI and Pexels database. All the software and data involved can be publicly available in the key resources table.

QUANTIFICATION AND STATISTICAL ANALYSIS

Adjacent pixel correlation analysis

The correlation coefficients are calculated as follows:

$$r_{xy} = \frac{\sum_{i=1}^{M}\left(x_i - \frac{1}{M}\sum_{j=1}^{M}x_j\right)\left(y_i - \frac{1}{M}\sum_{j=1}^{M}y_j\right)}{\sqrt{\sum_{i=1}^{M}\left(x_i - \frac{1}{M}\sum_{j=1}^{M}x_j\right)^2}\sqrt{\sum_{i=1}^{M}\left(y_i - \frac{1}{M}\sum_{j=1}^{M}y_j\right)^2}}$$

where x_i and y_i form the first pair of horizontal/vertical/diagonal/anti-diagonal adjacent pixels and M is the total number of horizontal/vertical/diagonal/anti-diagonal adjacent pixels.

Differential statistical analysis

Two standards are typically used to measure the dissimilarity between the source image and its encrypted version: the number of pixel change rate (NPCR) and the uniform average change intensity (UACI). In standard disparate attacks, attackers tend to make subtle modifications to the source image and then encrypt the original image using the proposed algorithm. This approach allows them to reveal the underlying relationship between the original and encrypted images. NPCR and UACI criteria are commonly used to evaluate the resistance of an encryption scheme to disparate attacks. The equations for calculating NPCR and UACI are as follows:

$$\begin{cases} NPCR = \frac{1}{H\times W}\times\sum_{i=1}^{H}\sum_{i=1}^{W}D(i,j)\times 100\% \\ UACI = \frac{1}{H\times W}\times\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{|v_1(i,j) - v_2(i,j)|}{255}\times 100\% \end{cases}$$

where $H\times W$ represents the size of the image, v_1, v_2 is the ciphertext image before and after changing one pixel of the plaintext image. D can be defined:

$$D = \begin{cases} 0, & if\ v_1(i,j) = v_2(i,j) \\ 1, & if\ v_0(i,j) \neq v_2(i,j) \end{cases}$$

Information entropy analysis

The degree of randomness in a system is typically assessed by using the entropy of the information as a standard metric. For an information source m, the information entropy $H(m)$ is given by:

$$H(m) = -\sum_{i=0}^{L}p(m_i)log_2p(m_i)$$

where L represents the total number of pixels. The probability of m_i is denoted by i$p(m)$.

Image quality analysis

In the realm of image processing, Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) are served as standard metrics for assessing the quality of encryption. The Mean Square Error (MSE) is part of the PSNR, defined as:

$$\begin{cases} MSE = \frac{1}{H \times W} \sum_{H}^{i=1} \sum_{W}^{j=1} (X(i,j) - Y(i,j))^2 \\ PSNR = 10 \times \log_{10}\left(\frac{Q^2}{MSE}\right) \end{cases}$$

where MSE represents the mean square error between the plaintext image X and the ciphertext image Y. The vertical extent of the image is represented by H, the horizontal dimension of the image is denoted by W. The pixel level of the image is denoted by Q. SSIM is a measure of the similarity between two images, explained as:

$$\mathrm{SSIM}(X,Y) = \frac{\left(2\mu_X\mu_Y + (0.01L)^2\right)\left(2\mu_{XY} + (0.03L)^2\right)}{\left(\mu_X^2 + \mu_Y^2 + (0.01L)^2\right)\left(\mu_X^2 + \mu_Y^2 + (0.03L)^2\right)}$$

where the mean values of the images X and Y are denoted by μ_X and μ_Y, respectively, while their respective standard deviations are represented as well. The dynamic range of pixel values is denoted as L.