

RESEARCH ARTICLE

A Dual-Domain Image Encryption Algorithm Based on Hyperchaos and Dynamic Wavelet Decomposition

QIUXIA QIN¹, ZHONGYUE LIANG¹, SHUANG LIU¹, XIAO WANG², AND CHANGJUN ZHOU^{1,3}¹College of Computer Science and Engineering, Dalian Minzu University, Dalian 116600, China²Xingzhi College, Zhejiang Normal University, Jinhua 321004, China³College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321000, China

Corresponding authors: Xiao Wang (tianzhu213@zjnu.cn) and Changjun Zhou (zhouchangjun@zjnu.edu.cn)


This work was supported in part by the National Natural Science Foundation of China under Grant 62272418 and Grant 62102058, and in part by the Basic Public Welfare Research Program of Zhejiang Province under Grant LGG18E050011.

ABSTRACT To ensure the safe and reliable transmission of images on public channels, this paper proposes a dual-domain image encryption algorithm based on hyperchaos and dynamic wavelet decomposition. The combination of dynamic wavelet decomposition and scrambling and diffusion operations is adopted in our algorithm to realize the combination of spatial and frequency domain encryption. This not only ensures the security of the encryption algorithm, but also ensures the robustness and operating efficiency of the encryption, and at the same time reduces the risk of being attacked. First, divide the original image into blocks, use a random number sequence to control the block scrambling process, and generate a scrambling matrix; Then by calculating the Hamming distance related to the plaintext, dynamically selecting the wavelet type, performing wavelet decomposition, and generating a wavelet coefficient matrix; Re-input the plaintext image to the SHA-512 algorithm to generate the initial value of the hyperchaos. The chaotic system generates the chaotic key matrix through iteration; Then the scrambling matrix is dynamically rotated, and then the Zigzag transform is used to generate the key matrix; Finally, the wavelet coefficient matrix, the chaotic key matrix, and the key matrix are subjected to bitwise XOR operation to realize the diffusion of pixel values and obtain the final encrypted image. Simulation experiments and performance analysis experiments can show that this algorithm can effectively encrypt and decrypt images, and has good encryption and decryption quality, and the ability to resist various attacks.

INDEX TERMS Dynamic wavelet decomposition, hyperchaos, image encryption, SHA-512, Zigzag transform.

I. INTRODUCTION

With the vigorous development of Internet technology, multimedia information such as text, voice, image, and video has been shared and disseminated on the Internet platform [1], [2]. Among them, image as an intuitive transmission medium has been widely used in the information age, and people use image data to convey information and exchange emotions [3], [4]. However, this form of communication also brings many risks. Once the necessary information of the image is maliciously obtained and tampered with by the attacker, it will

The associate editor coordinating the review of this manuscript and approving it for publication was Yeliz Karaca .

cause huge losses and harm to the user. Therefore, ensuring the safe sharing and dissemination of images on public channels has gradually become a vital topic in the field of information security, and has attracted more and more scholars' attention and research. Among them, encryption is one of the most common means of protecting information security. Compared with watermarking, hiding, steganography, etc. References [5], [6], [7], and [8], encryption can better protect the content of information, especially for the protection of private information.

Because chaos has inherent characteristics such as initial sensitivity, ergodicity, pseudo-randomness, etc. Reference [9], it is very in line with the requirements of

cryptography, and the chaotic system generates sequences very fast, so image encryption algorithms based on chaos theory have been continuously proposed and improved [10], [11], [12]. The control parameters of a chaos system are often used as the key of an encryption algorithm. In different chaos systems, the number of selected control parameters is different. Classical one-dimensional chaotic systems, such as Logistic chaotic map and Sine map [10], have disadvantages such as a few control parameters, small generated key space, limited chaotic interval, and low security. The image encryption algorithms based on chaos theory use the pseudo-random sequence generated by chaos to operate the image according to the characteristics of the periodic window. In addition, the image encryption effect is related to the periodic window. For example, logistic, sine, and tent chaotic maps [10] have short periodic windows, and it is difficult to guarantee the randomness of the encryption algorithm. In order to improve the security and randomness of encryption algorithms, and solve the shortcomings of low-dimensional chaos in encryption, researchers have proposed many high-dimensional chaotic systems and hyperchaos. For example, Xiu et al. [11] combined the memory characteristics of cellular neural networks to design a five-dimensional memory characteristic CNN hyperchaos, which has the characteristics of large parameter space and good chaotic characteristics. Zhu et al. [12] constructed a five-dimensional continuous hyperchaos. The chaos has two positive Lyapunov exponents, which is more random than general high-dimensional chaotic systems. They are used in encryption algorithms to improve the robustness of the algorithm and safety [12]. These encryption schemes incorporate many new technologies in the algorithm design, improve the complexity of the algorithm, and also ensure the security of ciphertext images. The above research results show that the chaotic system is a very useful tool for generating pseudo-random sequences for image encryption, and the chaotic-based image encryption scheme is an effective image encryption method. In addition, because hyperchaos have at least two positive Lyapunov exponents, they have stronger randomness. Compared with chaotic systems and high-dimensional chaotic systems, they have the characteristics of large key space, at least one control parameters, and large chaotic intervals. In recent years, it has received continuous attention and research from researchers, and has been widely used in chaotic image encryption.

Spatial domain encryption [13], [14], [15], [16], [17], [18], [19], [20] and frequency domain encryption [21], [22] provide different perspectives for image encryption. Between them, in the spatial domain encryption, it is mainly carried out by means of scrambling and diffusion. The scrambling operation changes the position of the pixel and breaks the strong correlation between the adjacent pixels of the image. The diffusion operation changes the size of the pixel value, and the size of the pixel value is mutually diffused, making the image more chaotic. Chaos image encryption algorithms based on spatial domain generally use a scrambling-diffusion framework, and use sequences generated by chaotic maps to

implement scrambling and diffusion operations. For example, Wang and Gao [23] proposed a chaos image encryption algorithm based on matrix semi-tensor product and compound key in 2020. This algorithm mainly performs semi-tensor product operations on chaotic sequences and scrambled images, which effectively improves the security of diffusion operations. In 2021, Wang et al. [24], [25] successively proposed two spatial image encryption algorithms. The first algorithm proposed [25] is an image encryption algorithm based on the chaotic diffusion value of the truth table. This algorithm uses the classic scrambling-diffusion framework to perform row, column, and diagonal bidirectional scrambling and diffusion operations by using nonlinear chaotic sequences, which greatly improve the effects of scrambling and diffusion, and the ciphertext image is simultaneously affected by the chaos system and the truth table rules, making it more secure. In the same year, the second algorithm proposed by Wang et al. [24] is an image encryption algorithm based on dynamic row scrambling and Zigzag transform. The algorithm is based on the idea of standard Zigzag scrambling, and the special traversal method was adopted to improve the scrambling effect. The results show that the encryption algorithm also has better security. In addition, in frequency domain encryption, the image is regarded as a two-dimensional signal with varying amplitude. The image can be processed by Fourier transform [21], discrete cosine transform [21], wavelet decomposition and other methods [22] to generate frequency coefficient matrix, the matrix is processed to a certain extent, and finally the corresponding inverse transformation can achieve the purpose of image encryption. For example, Wu et al. [26] proposed a new color image lossless encryption scheme based on two-dimensional discrete wavelet and six-dimensional hyperchaos. First, two-dimensional discrete wavelet is used to divide the image into four subbands, and then a constant factor is used to change the size of the subband, and finally the reconstruction of the subband. The experimental results show that the algorithm is safe, fast, and capable of resisting various attacks, which greatly enhances the performance of the encryption algorithm.

For the above two encryption methods, the spatial domain encryption is more secure, and the frequency domain encryption can make images more chaotic and more efficient. Both encryption methods can turn a visually meaningful image into a safe snowflake noise image, realizing effective image encryption. The above references all use a single domain to encrypt images. In order to further improve encryption efficiency and visual security, Wang et al. [27] proposed a three-image encryption and hiding algorithm based on chaos, compressed sensing and three-dimensional discrete cosine transform, which is based on dual-domain encryption, using two-dimensional discrete wavelet to transform the image into a sparse matrix, and then scrambling and compressing the sparse matrix twice, and finally embedding the matrix into the color carrier image. The results show the performance of the algorithm on statistical information such as correlation

coefficients and information entropy, all are better than single domain encryption. Hua et al. [28] proposed a new vision-safe image encryption scheme based on compressed sensing, which designed a new parallel compressed sensing technology and realized adaptive threshold sparsification using SWT. The results show that the algorithm has more high reconstruction quality and encryption. Yan et al. [29] proposed a chaotic image encryption algorithm based on fractional scrambling wavelet transform and three-dimensional cyclic shift operation. The original image is decomposed by three-level fractional wavelet to obtain low-frequency components and high-frequency components. The high-frequency components are scrambled by chaotic sequence, the low-frequency components are scrambled by three-dimensional cyclic shift, and finally reconstructed to obtain the encrypted image. Experimental results show that the algorithm has a good encryption effect. Shakir et al. [30] proposed an image encryption method based on the combination of Logistic chaotic mapping and wavelet transform. Firstly, Haar wavelet transform is performed on the plaintext image to obtain different frequency domains of the image, and then Logistic chaotic mapping is used to scramble the generated matrix, the test results show that the algorithm has achieved good encryption effect. It can be seen from the above analysis that the current research on dual-domain encryption schemes is neither sufficient nor perfect. Although combining the scrambling operation in the spatial with the frequency domain makes the encryption algorithm more resistant to attacks and has a better encryption effect, it ignores the diffusion operation that is as important as the scrambling operation in the spatial domain encryption. Therefore, the algorithm based on dual-domain encryption needs to be further improved.

Based on the above analysis, in order to further improve the encryption effect of dual-domain encryption, the frequency domain can not only be combined with the scrambling operation in the spatial domain, but also can be combined with the diffusion operation in the spatial domain, fully integrating the advantages of the two encryption methods. Therefore, this paper proposes a dual-domain image encryption algorithm based on hyperchaos and dynamic wavelet decomposition, which not only ensures the security but also takes into account the efficiency of encryption. The chaotic map selected in this paper is a hyperchaos, and the complex chaotic characteristics ensure the security of the algorithm. As for the architecture, the combination of scrambling and diffusion in the spatial domain and wavelet decomposition in the frequency domain improves the performance of the algorithm. In terms of efficiency, the improved Zigzag can achieve fast scrambling without increasing complexity. Additionally, the keystream used in the encryption process all relies on the original image. The experimental results compared with most existing encryption algorithms show that the proposed algorithm achieves the dual goals of security and efficiency.

The rest of the paper is organized as follows. Section II introduces the basic principles and knowledge of the algorithm. Section III introduces the specific encryption and

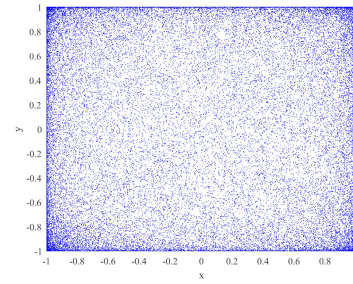


FIGURE 1. Bifurcation diagram of 2D-LSM.

decryption algorithms proposed. Section IV carries on the experiment simulation. Section V performs performance analysis. The last section concludes the paper.

II. RELATED THEORY OF THE ENCRYPTION ALGORITHM

A. 2D-LSM

The 2D-LSM [31] is obtained by diffusing the phase space of the one-dimensional Logistic map and the one-dimensional Sine map into two dimensions. The definition of this two-dimensional hyperchaos is as follows:

$$\begin{cases} x_{i+1} = \cos(4ax_i(1-x_i) + b\sin(\pi y_i) + 1); \\ y_{i+1} = \cos(4ay_i(1-y_i) + b\sin(\pi x_i) + 1). \end{cases} \quad (1)$$

where x, y are the system variables of the hyperchaotic system, a, b are the control parameters of the hyperchaos. Since the cosine transform is bounded for any input, the control parameters a, b can be arbitrarily large [31]. When $x(1) = 0.1, y(1) = 0.2, a = b = 60$, the resulting bifurcation diagram is shown in Fig. 1. Obviously, the trajectory of the 2D-LSM is distributed in the whole phase space, indicating that the chaotic system has complex chaotic characteristics. Besides, Fig. 2(a) gives the case of Lyapunov exponent (LE) when $a \in (1, 100), b = 60$, Fig. 2(b) shows the case of Lyapunov exponent (LE) when $a = 60, b \in (1, 100)$. It can be seen that 2D-LSM has a continuous chaotic range, the larger positive Lyapunov exponent, and is always in a hyperchaotic state.

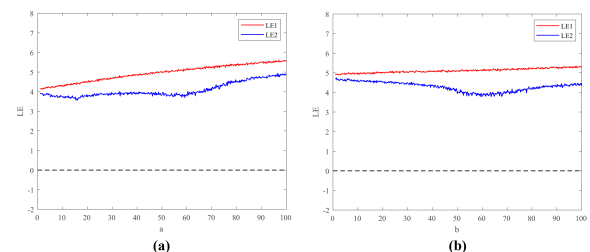


FIGURE 2. Lyapunov index spectrum of 2D-LSM (a) $LEs(b = 60)$, (b) $LEs(a = 60)$.

In this paper, the chaotic sequence generated by 2D-LSM is transformed into a chaotic key matrix to participate in the XOR diffusion process, which increases the randomness and security of the algorithm.

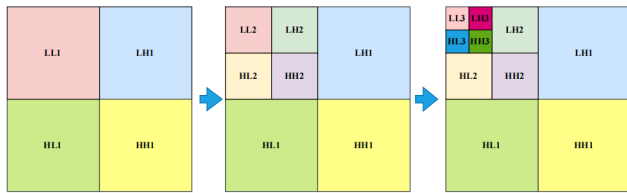


FIGURE 3. Three-layer wavelet decomposition diagram.

B. DYNAMIC WAVELET DECOMPOSITION

Wavelet decomposition is an important form of frequency domain encryption [27], [32]. In wavelet decomposition, the image is segmented through analysis filters. Each stage of the analysis filter is composed of high-pass and low-pass filters. After each level is filtered, an image subband will be generated. The generated image sub-bands include two types, namely low-pass sub-bands and high-pass sub-bands. Among them, there are three types of high-pass sub-bands, which are three high-pass sub-bands in the horizontal, vertical, and diagonal directions. Generally, LL represents the low-pass sub-band, LH represents the vertical high-pass sub-band, HL represents the horizontal high-pass sub-band, and HH represents the diagonal high-pass sub-band. If it is a multi-layer wavelet decomposition, the LL low-pass subband is decomposed in the same way. Until the end of multi-layer wavelet decomposition, a wavelet coefficient matrix is obtained. The following takes three-layer wavelet decomposition as an example to show the process of wavelet decomposition, as shown in Fig. 3.

Generally speaking, LL contains an approximate representation of the input image, while LH, HL, and HH only contain edge information and almost no energy. The process of three-layer wavelet decomposition can be clearly and intuitively seen from Figure 3. The LL of the upper level can be used as the input of the next level, and the relevant detail signals can be obtained through the filtering of each level. To increase the randomness of the algorithm, a dynamic wavelet decomposition method is proposed in this paper. This method uses the key related to the plaintext to select the specific wavelet type used in each wavelet decomposition, and calculates the specific number of wavelet decomposition layers according to the size of the input image matrix. Finally, the frequency coefficient matrix representing the original image is obtained.

C. ZIGZAG TRANSFORM

1) IMPROVED ZIGZAG TRANSFORM

The traditional Zigzag transform is often used to implement image scrambling due to its simplicity and ease of implementation and low time complexity [33]. Starting from the upper left corner of the image matrix, scan in a zigzag shape, put the scanned values into a one-dimensional sequence, and finally turn the one-dimensional sequence into a two-dimensional matrix to complete the scrambling process. The following takes a 3*3 matrix as an example to show the traditional Zigzag transform process, as shown in Fig. 4.

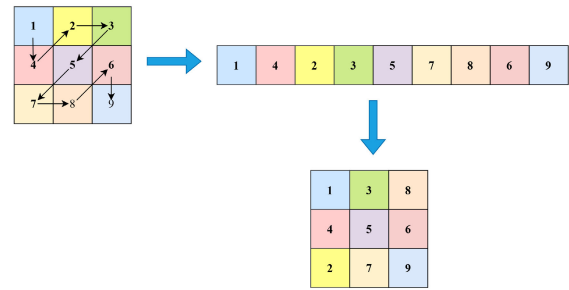


FIGURE 4. Traditional Zigzag transform.

It can be seen from Figure 4 that the pixel value of some positions in the scrambled matrix obtained by the traditional Zigzag transform of the 3*3 matrix is fixed, and the position of the pixel cannot be completely scrambled, and there are limitations in scanning. In addition, the traditional Zigzag transform works on a square matrix, which has a limited scope of application. In order to solve the above problems, this paper improves the traditional Zigzag transform to form an improved Zigzag transform. Firstly, the object of the improved Zigzag transform is no longer limited to square matrices, as shown in Fig. 5. Secondly, before the image matrix is scanned, the matrix is rotated first, and the specific rotation angle is determined by the key related to the plaintext, and then the matrix is transformed. This method makes the scanning result no longer single and fixed, and has stronger randomness, and does not greatly increase the computational complexity. Finally, when the traditional Zigzag transform transforms a square matrix with an odd side length, there is a problem that the most central element of the matrix cannot change its position after one round transform. Therefore, the improved Zigzag transform increases the backward shift processing of the center element and deal with this problem effectively. It should be noted that the improved Zigzag transform does not need to deal with non square matrices and even side length square matrices except for the special treatment of the central elements of odd side length square matrices. Taking a 3*3 square matrix as an example, an improved Zigzag transform is performed, as shown in Fig. 6.

Figure 5 shows the process of the improved Zigzag transform of non-square matrix, which shows that the improved Zigzag transform has a wider scope of application. In Figure 6, the generated Hamming distance related to the plaintext is used as a key to select a specific rotation angle, and then the matrix is rotated counterclockwise. The matrix rotation operation makes a matrix have four possible outputs. In addition, since the matrix is a square matrix with odd side lengths, the most central element of the output sequence is moved to the end of the sequence, and other elements remain unchanged, which solves the problem that some elements remain fixed. In this paper, the improved Zigzag transform is mainly used to generate the key matrix, and the generated key matrix will participate in the subsequent diffusion operation.

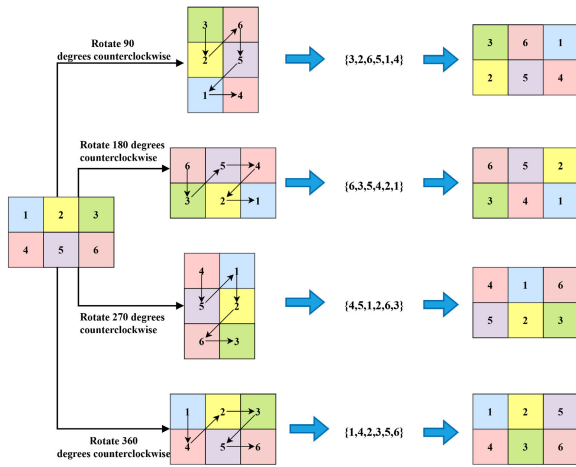


FIGURE 5. Improved Zigzag transform of non-square matrix.

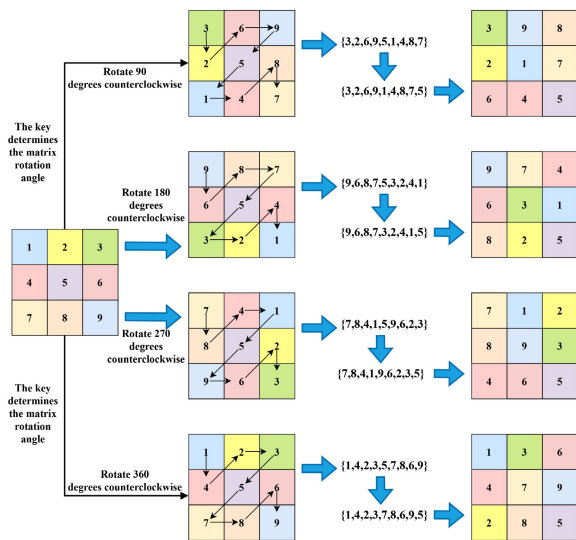


FIGURE 6. Improved Zigzag transform of odd-sided square matrix.

2) ZIGZAG PERFORMANCE EVALUATION

In addition to the problems of fixed elements and limited scope of use in traditional Zigzag transform, the generated scrambling matrix often has the problems of strong correlation between adjacent pixels and relatively concentrated distribution. In order to solve the above problems, Ramasamy et al. [34] also proposed an improved Zigzag transform, which firstly exchanges the first and second elements of the matrix with the last two elements, and then performs transformation, which distorts the relationship between image pixels. Fig. 7 shows the images obtained by performing three rounds transform on the Rice image by the traditional Zigzag transform, the Zigzag transform proposed by Ramasamy, and the improved Zigzag transform proposed in this paper. In addition, the correlation coefficients of adjacent pixels of each scrambled image in Figure 7 are also calculated using the Eq. (39), as shown in Table 1.

TABLE 1. Correlation coefficients of images in each round transform.

Round	Direction	Traditional Zigzag	Ramasamy-Zigzag [25]	Improved Zigzag
One round	Horizontal	0.0316	0.0363	0.0287
	Vertical	0.9355	0.9353	0.9356
	Diagonal	0.0307	0.0326	0.0257
Two rounds	Horizontal	0.0228	0.0278	0.0242
	Vertical	0.0436	0.0440	0.0390
	Diagonal	0.0197	0.0206	0.0190
Three rounds	Horizontal	0.0038	0.0046	0.0037
	Vertical	0.0127	0.0113	0.0036
	Diagonal	-0.0030	-0.0027	0.0017

It can be seen from Figure 7 that as the number of rounds increases, the images obtained by the three transform methods become more and more random, but the scrambling effect cannot be directly judged visually. It can be seen from the correlation coefficients in Table 1 that in one round transform, the correlation coefficients of the images generated by the improved Zigzag in some directions are better than other transform methods. As the number of rounds increases, the ability of the improved Zigzag transform to reduce the correlation becomes more and more prominent. Therefore, it can be shown that the improved Zigzag transform proposed in this paper can effectively reduce the correlation between adjacent pixels of the image.

III. PROPOSED IMAGE ENCRYPTION AND DECRYPTION ALGORITHM

This paper proposes a dual-domain image encryption algorithm based on hyperchaos and dynamic wavelet decomposition, which is mainly divided into two stages: scrambling and diffusion. It is worth mentioning that this algorithm uses frequency-domain wavelet decomposition for the diffusion process in the spatial domain. It is an effective fusion of spatial-domain encryption algorithm and frequency-domain encryption, which improves the security and real-time performance of encryption. In the scrambling stage, the image matrix is divided into blocks, and then a random number sequence is used to control the scrambling process of each block; In the diffusion stage, the scrambling matrix first generates a frequency coefficient matrix through dynamic wavelet decomposition, and then hyperchaos generates a chaotic key matrix, and then uses an improved Zigzag transform to generate a key matrix. Finally, the three kinds of matrices are XORed to complete the diffusion process in a simple and efficient way, and realize the encryption of the image. Section III-A-III-D will describe the encryption process in detail, and the encryption flow is shown in Fig. 8.

A. KEY GENERATION

The key of this algorithm mainly contains three parts: the initial value keys of the hyperchaos, the key used in the dynamic wavelet decomposition process to select the wavelet type, and the key used in the improved Zigzag transform to select the rotation angle of the image matrix. The specific process is shown in section III-A1-III-A3.

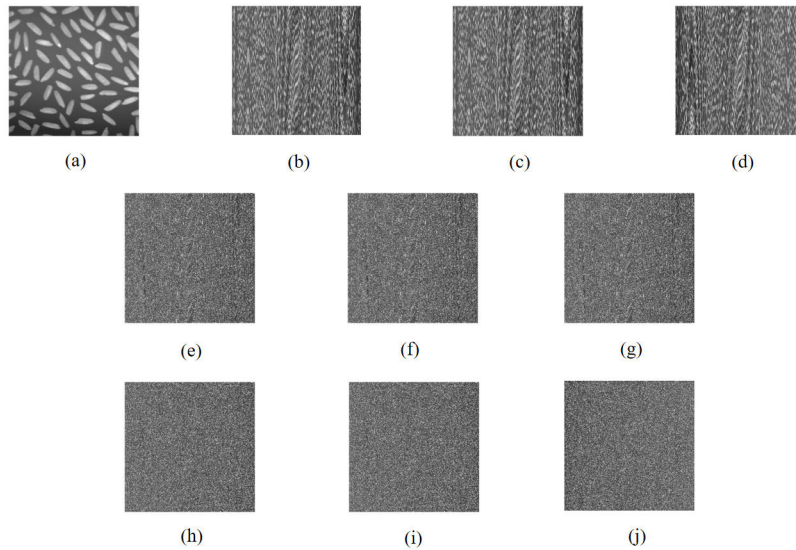


FIGURE 7. Image comparison of three rounds of transform of Rice image by three Zigzag transform methods (a) Rice original image, (b) traditional Zigzag one round transform, (c) Ramasamy-Zigzag one round transform, (d) improved Zigzag one round transform, (e) traditional Zigzag two rounds transform, (f) Ramasamy-Zigzag two rounds transform, (g) improved Zigzag two rounds transform, (h) traditional Zigzag three rounds transform, (i) Ramasamy-Zigzag three rounds transform, (j) improved Zigzag three rounds transform.

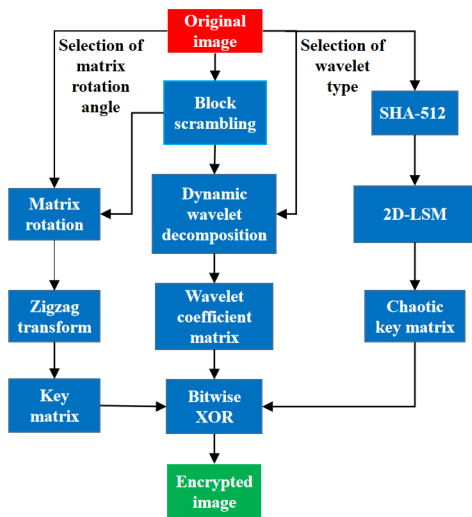


FIGURE 8. Encryption flowchart.

1) INITIAL VALUE KEY OF THE 2D-LSM

The biggest feature of the hash function is that the forward operation is relatively easy, and the reverse operation is very difficult. On the one hand, however large the input value is, it can be converted to 512 binary bits after using the SHA-512 function to output, and different inputs produce different results through this function, and the results are unique, so this part of the key is also unique. On the other hand, the keys generated by the hash function are related to the plaintext, which can effectively resist plaintext attacks. The initial value key of the 2D-LSM is generated by the original

image using the 512-bit message digest of the SHA-512 hash function. The specific process of the initial value key of the hyperchaos is as follows:

Step 1: The original image P is used as a variable of the hash function to generate a 512-bit message digest, as follows:

$$Info = hash(P, 'SHA - 512') \quad (2)$$

Among them, $hash()$ is the hash function, P is the original image, and $Info$ is the message digest generated.

Step 2: Since the generated message digest is given in the form of a 128-bit hexadecimal string, it is necessary to first convert the 128-bit hexadecimal string into 128 decimal, and further into 64 decimal $DInfo_1$. The process is as follows:

$$DInfo = hex2dec(Info(i)), i = 1, 2, \dots, 128 \quad (3)$$

$$DInfo_1 = DInfo(j) \times 10 + DInfo(j + 1), j = 1, 2, \dots, 64 \quad (4)$$

Among them, $hex2dec()$ is a function that transforms hexadecimal into decimal.

Step 3: Generate initial values $x(1), y(1)$ of 2D-LSM hyperchaotic system using 64 decimal numbers $DInfo_1$, take the value from the front to back interval, the interval is 4, a total of 17 values are generated, and the first 16 values are used to generate the initial chaotic value. The generation process is as follows:

$$x(1) = DInfo_1\{1\} \oplus DInfo_1\{4\} \oplus \dots \oplus DInfo_1\{28\} \quad (5)$$

$$y(1) = DInfo_1\{32\} \oplus DInfo_1\{36\} \oplus \dots \oplus DInfo_1\{60\} \quad (6)$$

Among them, \oplus is a bitwise XOR operation.

Step 4: There will be transient effects during the iteration of the 2D-LSM. To eliminate the transient effects, the random values generated by the previous iteration of the system need to be discarded. This algorithm uses the last decimal value of $DInfo_1$ to determine the specific iterative *discard* that needs to be discarded, as follows:

$$discard = 2000 + DInfo_1\{64\} \quad (7)$$

2) SELECTION KEY OF WAVELET TYPE

In the process of dynamic wavelet decomposition, which wavelet type is used for each image matrix is generated by the original image through the relevant theory of Hamming distance, as follows:

Step 1: The number of rows of the original image P is M and the number of columns is N , and a certain pixel value $P(i, j)$ of the original image is randomly selected using a random number generation function, as shown below:

$$\begin{cases} i = randperm(M, 1) \\ j = randperm(N, 1) \end{cases} \quad (8)$$

Among them, $randperm(n, k)$ means to generate k random numbers from $1 - n$.

Step 2: Change the pixel value $P(i, j)$ of the selected original image into 8-bit binary form, as follows:

$$P_1 = dec2bin(P(i, j), 8) \quad (9)$$

Among them, $dec2bin()$ is a function that converts a decimal number into a binary form. The second variable of the function represents the specific number of bits in the generated binary.

Step 3: The system gives an 8-bit binary sgn value, and calculates the Hamming distance HD between sgn and P_1 . Because the two variables are both 8-bit binary values, they are compared one by one starting from the first bit. If the two are equal, HD increases by 1. If the two are not equal, HD remains unchanged. After the comparison of the 8-bit values is completed, the final Hamming distance HD is obtained. Use the Hamming matrix HD to select one of the eight wavelet types, and continue the dynamic wavelet decomposition operation.

3) SELECTION KEY OF MATRIX ROTATION ANGLE

In the improved Zigzag transform, the matrix needs to be rotated first. The specific rotation angle is generated by the original image and the minimum concept, as shows:

Step 1: Calculate the minimum value of all pixel values of the original image P as follows:

$$Min = \min(\min(P)') \quad (10)$$

Among them, $\min()$ is a function to find the minimum value. Since P is a two-dimensional matrix, it is necessary to call the $\min()$ twice to calculate the minimum value of all elements.

Step 2: Since there are 4 possible rotation angles, it is necessary to perform certain processing on the Min value

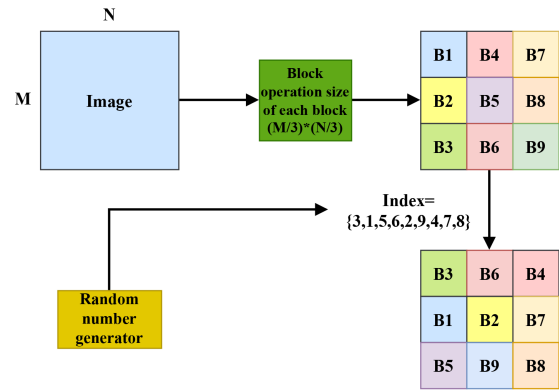


FIGURE 9. Schematic diagram of block scrambling.

so that the value of the generated selection angle $RotAng$ is between 1 and 4. The processing process is as follows:

$$RotAng = \text{mod}(Min, 4) + 1 \quad (11)$$

Among them, $\text{mod}()$ is the remainder function.

B. BLOCK SCRAMBLING STAGE

This stage is mainly to divide the original image P into blocks, and then generate a random number sequence, use the random number sequence to scramble each block, and complete the process of generating the matrix $P_scramble$ after the scrambling. The principle of block scrambling is shown in Fig. 9. Among them, the “Index={3,1,5,6,2,9,4,7,8}” in Fig. 9 is a pseudo-random number sequence obtained by a random number generator, and the pseudo-random number sequence is used to control block scrambling. The specific process is as follows:

Step 1: The plaintext image P is divided into 16×16 small blocks, the specific process is as follows:

$$\begin{cases} r = \text{size}(P, 1)/16 \\ c = \text{size}(P, 2)/16 \end{cases} \quad (12)$$

$$P_scramble = \text{mat2cell}(P, \text{ones}(1, r) \times 16, \text{ones}(1, c) \times 16) \quad (13)$$

Among them, $\text{size}()$ is a function to calculate the size of the matrix. When the second variable of the function is 1, the output result is the total number of rows in the matrix. When the second variable of the function is 2, the output result is the total number of columns in the matrix. The $\text{mat2cell}()$ can divide a matrix into blocks of different sizes. $\text{ones}()$ is a function to generate a matrix of all ones.

Step 2: Use the random number generation function to generate a random number sequence with the same size as the total block number $bsum$, the generation process is as follows:

$$bsum = r \times c \quad (14)$$

$$\text{seq_scramble} = \text{randperm}(bsum) \quad (15)$$

Among them, $randperm(n, k)$ means to generate k random numbers from $1 - n$.

Step 3: Use the *seq_scramble* sequence to perform block scrambling on each block, and obtain the scrambling matrix after completion. The process is as follows:

$$P_scramble = cell2mat(reshape(P_scramble \\ (seq_scramble), r, c)) \quad (16)$$

Among them, *cell2mat()* converts a cell array composed of multiple matrices into a matrix function, and *reshape()* is a function to transform a specified matrix into a matrix of specific dimensions.

C. DIFFUSION STAGE

The wavelet coefficient matrix generated by dynamic wavelet decomposition, the chaotic matrix generated by the 2D-LSM, and the key matrix generated by the improved Zigzag transform are sequentially XORed to realize the diffusion process. Although the diffusion process is simple and easy to achieve, the diffusion effect is good. The details are shown in section III-C1-III-C4.

1) DYNAMIC WAVELET DECOMPOSITION

Select the number of wavelet decomposition layers according to the size of the original image *P*, use the *HD* key generated in section III-A2 to select the specific wavelet type used for each encryption, and then perform wavelet decomposition on the matrix generated after the scrambling, and the decomposition is completed to generate a wavelet coefficient matrix *P_W*, details as follows:

Step 1: Calculate the specific level *level* of wavelet decomposition according to the original image *P*, the process is as follows:

$$n = size(P, 1) \quad (17)$$

$$level = \log_2(n) \quad (18)$$

Among them, the *size()* function is used to calculate the size of the matrix, and *log₂()* is a logarithmic function.

Step 2: According to the key *HD*, select the specific wavelet *Wavelet* to be used, and calculate the four filters associated with this type of wavelet. The process is as follows:

$$[L_D, H_D, L_R, H_R] = wfilters(Wavelet) \quad (19)$$

Among them, *wfilters()* is the function used to calculate the wavelet filter, *L_D* is the decomposition low-pass filter, *H_D* is the decomposition high-pass filter, *L_R* is the reconstruction low-pass filter, and *H_R* is the reconstruction high-pass filter.

Step 3: The scrambled matrix *P_scramble* and *L_D* are the decomposition low-pass filter, *H_D* is the decomposition high-pass filter, and the wavelet decomposition layer number *level* is used as the variable parameter of the wavelet decomposition to perform the multi-layer wavelet decomposition operation. After the multi-layer decomposition is completed, the frequency wavelet coefficient matrix *P_W* is generated.

2) GENERATING CHAOTIC MATRIX OF 2D-LSM

Using the initial value of the 2D-LSM generated in section III-A1, the 2D-LSM is iterated many times to generate the chaotic key matrix *CM*, details as follows:

Step 1: The length of the original image *P* is *M*, the width is *N*, and *discard* is the number of iterations that need to be discarded in the early stage of the chaotic system generated in section III-A1. First, determine the total number of iterations of the chaos. The process is as follows:

$$sum_discard = ceil(M \times N/2) + discard \quad (20)$$

where *ceil()* is the round-up function.

Step 2: In order to avoid transient effects, using the initial chaos value in III-A1, the 2D-LSM is iterated for *discard* times, and the generated chaotic value is discarded.

Step 3: Then continue to iterate. Since the chaotic system is two-dimensional, each iteration will generate two chaotic random values. The two chaotic random values generated each time are placed in sequence *seq₁* until the number of iterations reaches *sum_discard*, generating *M*N* size chaotic sequence *seq₁*.

Step 4: Perform certain processing on the chaotic sequence *seq₁*, first control the value of the chaotic sequence between 1-255, and then convert the chaotic sequence into a chaotic matrix *CM* a length of *M* and a width of *N*, details as follows:

$$seq_1 = mod(floor(seq_1 \times 10^{10}), 256) \quad (21)$$

$$CM = reshape(seq_1, M, N) \quad (22)$$

where *floor()* is a round down function, *mod()* is a remainder function, and the *reshape()* is a matrix reconstruction function.

3) IMPROVED ZIGZAG TRANSFORM TO GENERATE KEY MATRIX

The improved Zigzag transform consists of two parts. First, use the *RotAng* key generated in section III-A3 to control the angle of the matrix rotation and perform the rotation of the matrix. Then, the rotated matrix is scanned by Zigzag to generate a one-dimensional sequence, and the one-dimensional sequence is transformed into a two-dimensional matrix to obtain the key matrix *KM*, details as follows:

Step 1: Rotate the scrambled matrix *P_scramble* according to *RotAng*, as follows:

$$RM = rot90(P_scramble, RotAng) \quad (23)$$

Among them, the *rot90()* is a function that makes a matrix rotate counterclockwise.

Step 2: Perform a Zigzag scan on the matrix *RM* after matrix rotation to generate a one-dimensional sequence *seq₂*, as follows:

$$seq_2 = zigzag(RM) \quad (24)$$

Among them, the *zigzag()* is a function for performing Zigzag scanning.

Step 3: Turn the one-dimensional sequence seq_2 into a key matrix KM of size $M * N$, as follows:

$$KM = reshape(seq_2, M, N) \quad (25)$$

Among them, the $reshpae()$ is a matrix reconstruction function.

4) SPECIFIC DIFFUSION PROCESS

The wavelet coefficient matrix P_W generated in section III-C1, the 2-D chaotic key matrix CM generated in section III-C2, and the key matrix KM generated in section III-C3 are XORed to achieve diffusion, and the final encrypted image C is obtained, details as follows:

Step 1: First, process the P_W matrix and control the value of the matrix between 1-255. The process is as follows:

$$P_W = mod(floor(P_W \times 10^{15}), 256) \quad (26)$$

where $floor()$ is a round down function, and $mod()$ is a remainder function.

Step 2: The P_W matrix, the CM matrix and the KM matrix are bitwise XORed to obtain the encrypted matrix C , as follows:

$$C(i, j) = P_W(i, j) \oplus CM(i, j) \oplus KM(i, j) \quad (27)$$

$$i = 1, 2, \dots, M \quad (28)$$

$$j = 1, 2, \dots, N \quad (29)$$

Among them, \oplus is a bitwise XOR operation.

D. ENCRYPTION ALGORITHM

The dual-domain image encryption algorithm proposed in this paper based on hyperchaos and dynamic wavelet decomposition, the specific steps are as follows:

1) The original image P is divided into blocks first, and then each block is scrambled under the control of a random number sequence, as shown in section III-B.

2) The key obtained in section III-A2 is used to select the wavelet type, and then the dynamic wavelet decomposition operation is performed on the matrix $P_scramble$ obtained after the scrambling to obtain the wavelet coefficient matrix P_W , as shown in section III-C1.

3) In section III-A1, the initial value key of the 2D-LSM is obtained as the initial value of chaos, and the chaos is iterated. In order to avoid transient effects, the number of discarded previous iterations and the total number of iterations are calculated in advance. When all the system iterations are over, the chaotic key matrix CM is generated, as shown in section III-C2.

4) The Hamming distance HD key obtained in section III-A3 is used to select the angle of matrix rotation. The matrix $P_scramble$ after the scrambling is first subjected to matrix rotation, and then Zigzag transform is performed to generate the key matrix KM , as shown in section III-C3.

5) The P_W matrix, the KM matrix and the CM matrix are sequentially subjected to the bitwise XOR operation to realize the diffusion operation, complete the image encryption, and obtain the encrypted image C , as shown in section III-C4.

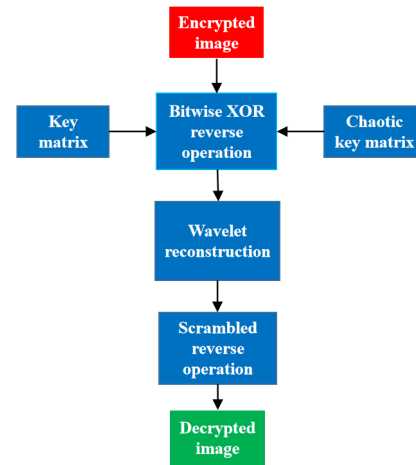


FIGURE 10. Decryption flowchart.

E. DECRYPTION ALGORITHM

The decryption algorithm and the encryption algorithm have a reversible relationship [15], [17], [20]. Among them, the encryption key and the decryption key are consistent, and the chaotic sequence used in the decryption process is also consistent with the encryption algorithm. The decryption party uses the decryption key to generate the chaotic key matrix and the key matrix according to section III-C2 and section III-C3. Also note the following: First of all, the purpose of the scrambling operation in the encryption and decryption algorithm is to disturb the position of the pixel value and break the strong correlation between the pixels. This is a reversible process. Second, the purpose of the diffusion operation in the encryption and decryption algorithm is to change the size of the pixel value, and this process is also reversible. In addition, the wavelet decomposition operation and wavelet reconstruction operation in the frequency domain are also reversible. Therefore, the decryption algorithm in this paper is mainly divided into three parts: the inverse process of the diffusion operation, the wavelet reconstruction process, and the inverse process of the scrambling operation. The flowchart of the entire decryption algorithm is shown in Fig. 10.

The concrete steps of the decryption algorithm are as follows:

1) The encrypted image C , the key matrix KM , and the key chaotic matrix CM are sequentially performed the inverse operation of the bitwise XOR operation to obtain the wavelet coefficient matrix C_W .

2) Input the wavelet coefficient matrix C_W , L_R reconstruction low-pass filter, H_R reconstruction high-pass filter, and wavelet decomposition layer level $level$ as parameters into the wavelet reconstruction function, reconstruct the wavelet coefficient matrix, and get the result after scrambling matrix $C_scramble$.

3) The matrix $C_scramble$ obtained in 2) is divided into blocks, and the process is as follows:

$$C_scramble = mat2cell(C_scramble, ones(1, r) \times 16, ones(1, c) \times 16) \quad (30)$$

TABLE 2. Key table.

Key type	Key
2D-LSM initial value key	$x(1)=0.4688, y(1)=0.6250$
2D-LSM discard key	$discard=2065;$
Matrix rotation angle selection key	$RotAng=1;$
Wavelet type selection key	$HD='10001111';$
Hamming distance key	$sgn='11111111';$

Among them, $mat2cell()$ is a function to transform a matrix into a cell array, and $ones()$ is a function to generate a matrix of all ones.

4) Continue to reverse scrambling each block of $C_scramble$ to obtain the decrypted image D . The specific process is as follows:

$$iseq_1 = 1 : length(seq_1) \quad (31)$$

$$seq = [seq'_1, iseq'_1] \quad (32)$$

$$seq = sortrows(seq, 1) \quad (33)$$

$$D = cell2mat(reshape(C_scramble(seq(:, 2)), r, c)) \quad (34)$$

Among them, seq_1 is the random number sequence used when the encryption algorithm is scrambling, $length()$ is the function to calculate the length, $sortrows()$ is the function to sort the matrix according to the specified row, $reshape()$ is the matrix reconstruction function, $cell2mat()$ is a function that converts a cell array composed of multiple matrices into a matrix.

IV. EXPERIMENTAL SIMULATION

This section shows the experimental simulation results of image encryption and decryption. The specific operating environment is: personal notebook computer, window10 operating system, Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50GHz, 16G memory, Matlab2020a simulation platform. The keys used in this simulation experiment are shown in Table 2.

The experimental simulation in this section is mainly divided into three parts: the encryption and decryption experiments on four groups of 512*512 images with different contents, and the experiments on 256*256, 512*512, 1024*1024 images with different sizes and different contents to perform image encryption and decryption experiments. Perform image encryption and decryption experiments on all black and white images. Experimental images are from <https://unsplash.com>. Fig. 11 shows the experimental results of encryption and decryption of four groups of 512*512 images.

As can be seen from Fig. 11, 512*512 images can be effectively encrypted and decrypted. Fig. 12 shows the experimental results of the encryption and decryption experiment of the algorithm on three groups of images of different sizes.

It can be seen from Fig. 12 that the size of the image does not affect the normal encryption and decryption of this

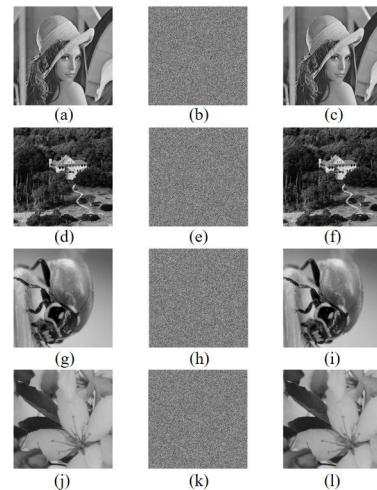


FIGURE 11. 512*512 size image encryption and decryption experiment results (a) Lena_512 original image, (b) Lena_512 encrypted image, (c) Lena_512 decrypted image, (d) Building original image, (e) Building encrypted image, (f) Building decrypted image, (g) Insect original image, (h) Insect encrypted image, (i) Insect decrypted image, (j) Flower original image, (k) Flower encrypted image, (l) Flower decrypted image.

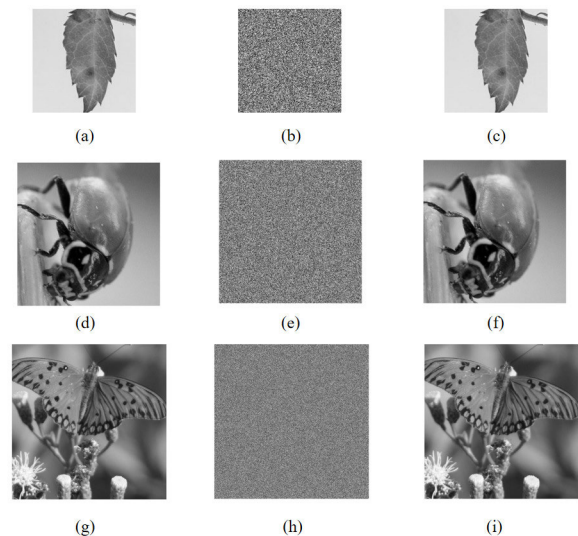


FIGURE 12. Experimental results of encryption and decryption of images of different sizes (a) Leaf original image, (b) Leaf encrypted image, (c) Leaf decrypted image, (d) Insect original image, (e) Insect encrypted image, (f) Insect decrypted image, (g) Butterfly original image, (h) Butterfly encrypted image, (i) Butterfly decrypted image.

algorithm. Fig. 13 shows the experimental results of using White and Black images for encryption and decryption.

It can be seen from Fig. 13 that even if a all white image or a all black image is used, secure and effective encryption and decryption can still be achieved, indicating that the algorithm has a strong ability to resist malicious attacks. From the experimental results, all the images after encryption become noise-like images, which can hide the effective information of the plaintext image well and meet the requirements of image encryption. And all the images can get the correct

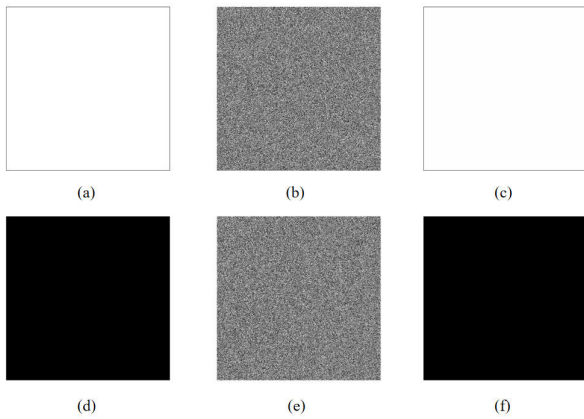


FIGURE 13. Experimental results of encryption and decryption of all white and all black images (a) White original image, (b) White encrypted image, (c) White decrypted image, (d) Black original image, (e) Black encrypted image, (f) Black decrypted image.

decryption effect, which further proves the effectiveness of this algorithm. In short, the encryption algorithm proposed in this paper can encrypt and decrypt images of different sizes, or all black and all white images, and has a wide range of applications and has certain practical application value.

V. PERFORMANCE ANALYSIS

Security analysis is the most important performance indicator of encryption algorithm [34], [35], [36]. Therefore, this section evaluates the security of encryption algorithms in key space and sensitivity, security analysis of chaotic key generation functions, and statistical analysis. In addition, a good encryption algorithm must be able to resist all known types of attacks, so this section analyzes the ability of the proposed algorithm to resist all known attacks. In addition, a good encryption algorithm must have a certain timeliness and can accomplish the effect of fast encryption and decryption in a short enough time, so this section gives an analysis of time efficiency.

A. KEY SPACE ANALYSIS

The key space represents the sum of various keys used in the encryption system, and the size of the key space can measure the ability of the encryption algorithm to resist brute force attacks [37], [38]. When the key space of the encryption algorithm is greater than 2^{100} , the security of the encryption system can be proved.

The key space of this algorithm will be analyzed from two aspects below. On the one hand, the key of this algorithm mainly includes two initial value keys of 2D-LSM, the number of iterations discarded by chaotic system, the selection key of matrix rotation angle in improved Zigzag transform, the selection key of wavelet type, and the given key of the system when calculating Hamming distance. Assuming that the calculation accuracy of the double-precision value is 10^{-16} , the key space of this encryption algorithm is $(10^{16})^4 + (2^8)^2$, which is much larger than the required 2^{100} . On the

other hand, this algorithm uses the SHA-512 hash algorithm to generate a key. In this process, the hash algorithm can generate a 512-bit message digest, representing a key space of at least 2^{512} , which is far greater than 2^{100} . In short, the key space of this algorithm is large enough to resist brute force attacks.

B. KEY SENSITIVITY ANALYSIS

Key sensitivity analysis is the analysis of the sensitivity of the image key, and is an important indicator for evaluating the security of the encryption system [39], [40]. Key sensitivity mainly includes encryption key sensitivity and decryption key sensitivity. In a secure encryption and decryption algorithm, when the encryption key is slightly changed, the corresponding encrypted image will change to a greater extent; When the decryption key is slightly changed, the corresponding decrypted image will also change to a larger degree. The greater the degree of change in the corresponding encrypted image or decrypted image, the stronger the key sensitivity.

The pixel change rate NPCR and the uniform average change intensity UACI are usually used to measure the sensitivity of the key [40]. Among them, the ideal value of NPCR is 99.6054%, and the ideal value of UACI is 33.4653% [41]. The stronger the key sensitivity is, the closer the two index values obtained are to the ideal value. The definitions of NPCR and UACI are as follows:

$$NPCR(D_1, D_2) = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} C(i, j)}{M \times N} \times 100\% \quad (35)$$

$$UACI(D_1, D_2) = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \frac{|D_1(i, j) - D_2(i, j)|}{255}}{M \times N} \times 100\% \quad (36)$$

$$C(i, j) = \begin{cases} 0, & D_1(i, j) = D_2(i, j) \\ 1, & D_1(i, j) \neq D_2(i, j) \end{cases} \quad (37)$$

Among them, D_1 and D_2 are the two encrypted images before and after changing the key respectively. Their length is M and their width is N .

This experiment uses the 2D-LSM initial value key $x(0)$ as the key to change. The specific key change method: $x(0) = 0.4688$ becomes $x(0) = 0.4688 + 10^{-16}$. Perform NPCR and UACI calculations on the encrypted images of different sizes and different contents and the encrypted images generated after the keys are changed. The NPCR and UACI values obtained are shown in Table 3.

Table 3 shows the NPCR and UACI values of the encrypted images before and after the image key changes of different sizes and different contents. Through numerical comparison, it can be seen that the seven groups of values are very close to the ideal value. Therefore, the quantitative analysis of the formula can show that the key sensitivity of this algorithm is strong. In addition to using data to illustrate the degree of difference between encrypted images obtained by changing the key, in addition to the Lena_512 image as an example, this section gives an intuitive image comparison situation. Fig. 14 shows the encrypted image before and after the key is

TABLE 3. NPCR, UACI table.

Image name	Image size	NPCR(%)	UACI(%)
Lena_256	256*256	99.6056	33.4650
Leaf	256*256	99.6052	33.4657
Lena_512	512*512	99.5994	33.4592
Building	512*512	99.6007	33.4709
Insect	512*512	99.5998	33.4607
Flower	512*512	99.6002	33.4596
Butterfly	1024*1024	99.6109	33.4737

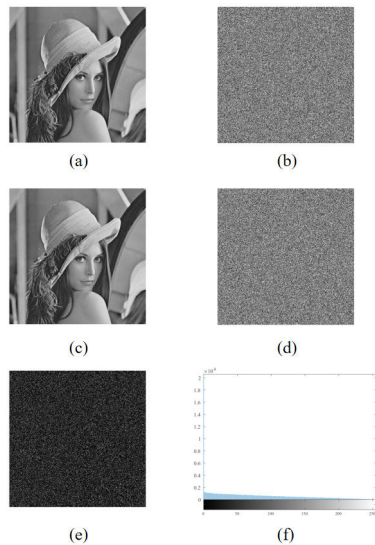


FIGURE 14. Simulation results of key sensitivity (a) Lena_512 original image, (b) Lena_512 encrypted image, (c) Lena_512 original image, (d) Lena_512 encrypted image with a key change of one bit, (e) difference image of two encrypted images, (f) histogram of the difference image.

changed, the difference image of the two encrypted images, and the histogram of the difference image.

As can be seen from the above images, although the two encrypted images are both in the shape of snowflake noise, the difference between the two cannot be seen by the naked eye, but the difference image of the two encrypted images and the histogram of the difference image can be proved that there is indeed quite different between the two encrypted images, so the image comparison can prove that the key sensitivity of this algorithm is strong. In short, whether it is through numerical quantitative analysis or image comparison, it can be proved that this algorithm is very sensitive to keys and has high security.

C. SECURITY ANALYSIS OF THE CHAOTIC KEY GENERATION FUNCTION

Randomness is a vital indicator to measure a encryption system [42]. When the encryption system does not meet the requirements of randomness, the security of the system will be threatened, affecting the normal use of the system.

In an encryption system, there are four basic requirements for randomness. Among them, the first requirement is that

TABLE 4. Information entropy table.

Algorithm	Image name	Image size	Information entropy
Proposed	Lena_256	256*256	7.9989
Proposed	Lena_512	512*512	7.9994
Proposed	Butterfly	1024*1024	7.9998
Proposed	Black	512*512	7.9992
Proposed	White	512*512	7.9992

the generated sequence needs to have strong statistical characteristics. In this paper, a 2D-LSM is used in the encryption system to generate a pseudo-random sequence, the test results show that the entropy source has strong statistical characteristics. The second requirement of encryption randomness generates a subset of random numbers that cannot be used to predict other random numbers. The third requirement and the fourth requirement are specific manifestations of the second requirement. As we all know, the one-way function is very suitable for this requirement. One of the uniqueness of the encryption algorithm proposed in this paper is the use of the proven secure encryption primitive SHA-512 algorithm, which can provide a single functional requirement. In other words, provided the SHA-512 algorithm is safe, the randomness of this algorithm can be guaranteed.

D. INFORMATION ENTROPY ANALYSIS

Information entropy is an indicator used to describe the randomness of disordered information. Through information entropy, the level of confusion in pixel distribution can be indicated [7]. For a closed system, information entropy is a finite number. The value of information entropy in an ideal state is 8 [43], and the mathematical definition formula of information entropy is as follows:

$$H(m) = - \sum_{i=0}^{i=2^M-1} P(m_i) \log_2 P(m_i) \quad (38)$$

Among them, m_i represents the i -th gray value of the image m , M represents the number of binary digits required to represent the gray value m_i , and $P(m_i)$ is the probability of m_i appearing. Under normal circumstances, the closer the information entropy is to 8, the stronger the ability of the encrypted image to hide the original image information, and the higher the security of the algorithm. Table 4 shows the information entropy of images with different content and different sizes after being encrypted by this algorithm and Table 5 shows the comparison with the information entropy of the algorithm [38], [42], [43], [44], [45], [46].

It can be seen from Table 4 that the information entropy value of the ciphertext image get from images of different sizes and different contents or all black and all white images are very approach to the ideal value of 8, which means that the ciphertext image get after the original image is encrypted by the encryption algorithm hardly reveals any useful information. In addition, compared with the information entropy values of references in Table 5, the information entropy values of

TABLE 5. Lena information entropy comparison.

Algorithm	Image name	Information entropy
Proposed	Lena_256	7.9989
Ref. [44]	Lena_256	7.9983
Ref. [45]	Lena_256	7.9989
Ref. [46]	Lena_256	7.9982
Proposed	Lena_512	7.9994
Ref. [38]	Lena_512	7.9993
Ref. [42]	Lena_512	7.9916
Ref. [43]	Lena_512	7.9994

this algorithm are slightly above than that of other references, and are closer to the ideal value. Therefore, the algorithm can effectively turn the image into a chaotic state, and the security is very high.

E. STATISTICAL ANALYSIS

The experiment in this section mainly includes two aspects: correlation statistical analysis and histogram statistical analysis. Correlation analysis is used to analyze the degree of correlation between adjacent pixels in plaintext images and adjacent pixels in ciphertext images, and to determine whether the proposed encryption algorithm has the ability to eliminate strong correlation [9]. The histogram statistical analysis is to analyze the distribution of the pixel values of the original image and the encrypted image, by comparing the original image and the encrypted image histogram to determine whether the proposed algorithm can make the histogram pixel values evenly distributed [18]. The details are shown in section V-E1-V-E2.

1) CORRELATION ANALYSIS

The correlation coefficient is an index used to evaluate the degree of correlation between adjacent pixels [21]. The correlation of adjacent pixels includes horizontal correlation, vertical correlation, and diagonal correlation. For plaintext images, the correlation between adjacent pixels is very strong, and the attacker can decrypt the ciphertext image through statistical attacks. However, this strong correlation is not obvious in the ciphertext image, and the correlation of the encrypted image is low, which can effectively crack the statistical attack. The correlation of adjacent pixels is defined as follows:

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{39}$$

$$cov(x, y) = \frac{1}{N} \times \sum_{i=1}^N (x_i - E(x))(y_i - E(y)) \tag{40}$$

$$D(x) = \frac{1}{N} \times \sum_{i=1}^N (x_i - E(x))^2 \tag{41}$$

$$E(x) = \frac{1}{N} \times \sum_{i=1}^N x_i \tag{42}$$

TABLE 6. Correlation coefficient table.

Image	Original image			Encrypted image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena_256	0.9414	0.9723	0.9188	0.0038	0.0002	-0.0012
Lena_512	0.9713	0.9849	0.9588	0.0004	0.0026	-0.0015
Butterfly	0.9961	0.9955	0.9922	0.0002	-0.0020	0.0002
Black	NaN	NaN	NaN	-0.0011	-0.0024	0.0007
White	NaN	NaN	NaN	-0.0022	0.0023	0.0011

TABLE 7. Lena_256 correlation coefficients comparison.

Direction	Lena_256	Ref. [44]	Ref. [45]	Ref. [46]
Horizontal	0.0038	-0.0050	0.0116	-0.0015
Vertical	0.0002	0.0006	0.0126	-0.0032
Diagonal	-0.0012	0.0015	0.0218	0.0023

TABLE 8. Lena_512 correlation coefficients comparison.

Direction	Lena_512	Ref. [38]	Ref. [42]	Ref. [43]
Horizontal	0.0004	-0.0008	-0.0074	0.0032
Vertical	0.0026	0.0010	0.0019	0.0021
Diagonal	-0.0015	0.0026	-0.0043	0.0082

Among them, x and y are two adjacent pixels, N represents the total logarithm of randomly selected adjacent pixels, $cov(x, y)$ represents covariance, $D(x)$ represents variance, and $E(x)$ represents expectation. This paper selects 1500 pairs of adjacent pixels in the horizontal, vertical, and diagonal directions of the original image and encrypted image of different sizes and all black and all white images, and calculates the correlation coefficient by eq. (39). The obtained correlation coefficient is shown in Table 6. In addition, Table 7 and Table 8 show the comparison with related references [38], [42], [43], [44], [45], [46].

It can be seen from Table 6 that no matter what type of image it is, the correlation coefficient of adjacent pixels before encryption is approach to 1, and the correlation is very strong. However, the correlation coefficient between adjacent pixels of the encrypted image is approach to 0, and the correlation is significantly reduced. Moreover, Table 7 and Table 8 show the comparison with the latest references and also show that the algorithm has superior performance in reducing the correlation of adjacent pixels, can resist statistical attacks very effectively, and ensure the security of the algorithm.

The points with the same or similar pixel values in the original image will be distributed in a concentrated manner. This situation is most obvious in the three directions of horizontal, vertical, and diagonal, while the pixels of the encrypted image will be uniformly distributed. This section draws the correlation images of adjacent pixels in the three directions of the Lena_512 image, and intuitively shows the distribution of adjacent pixels through the images. The specific situation is shown in Fig. 15.

It can be seen from Fig. 15 that the pixels in the three directions of the original image are concentrated in the diagonal

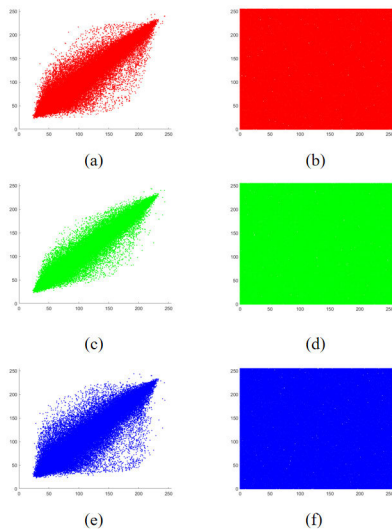


FIGURE 15. Correlation scatter diagram in three directions (a) original image horizontal direction, (b) encrypted image horizontal direction, (c) original image vertical direction, (d) encrypted image vertical direction, (e) original image diagonal direction, (f) encrypted image diagonal direction.

and the areas on both sides of the diagonal, indicating that there is a strong correlation. However, the pixels in the three directions of the encrypted image are evenly distributed within the coordinate range, indicating that the correlation is significantly weakened. Therefore, both quantitative analysis and qualitative analysis show that this algorithm has the ability to reduce the correlation of encrypted images and resist statistical attacks.

2) HISTOGRAM ANALYSIS

The histogram of an image is used to describe the distribution of image pixel values. It is generally used to indicate the number of pixels with the same gray value or the frequency of the pixel [6]. The abscissa of the histogram represents the gray level of the pixel, and the ordinate is used to represent the number of pixels that appear in the gray level [46]. Normally, natural images reflect the real and observable world, so the histogram distribution of natural images presents a inhomogeneous state. But for the image obtained after encryption, since the visual requirements must be chaotic, the histogram distribution should show a uniform state, so as to ensure that no effective information can be get from the encrypted image. Fig. 16 shows the original image, encrypted image, decrypted image and their respective histograms using the Lena_512 image as an example.

As can be seen from the above images, the histogram of the original image is distributed in the middle range, with obvious statistical law, and the distribution of pixel values fluctuates; The histogram of the encrypted image shows an approximately uniform distribution, which is obviously different from the histogram of the original image. Therefore, it can be shown that this algorithm can make the histogram

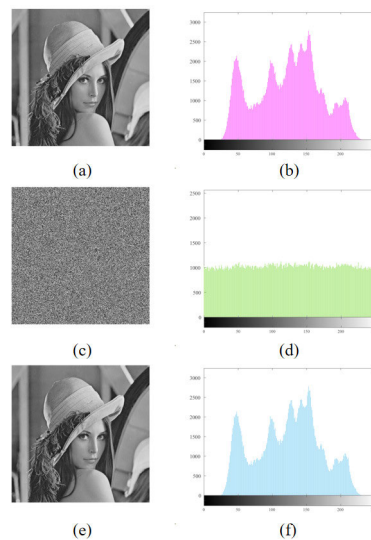


FIGURE 16. Histogram of Lena_512 image (a) original image, (b) original image histogram, (c) encrypted image, (d) encrypted image histogram, (e) decrypted image, (f) decrypted image histogram.

distribution of encrypted images more uniform and has the ability to resist statistical attacks.

F. PSNR ANALYSIS

PSNR is an objective index to measure whether there is distortion in an image [43]. In order to quantify the visual quality of the decrypted image in quantitative form and compare the difference between the decrypted image and the original image, it is usually measured by PSNR. Its mathematical definition is as follows:

$$PSNR = 10 \times \log_{10}\left(\frac{MAX_D^2}{MSE}\right) = 20 \times \log_{10}\left(\frac{MAX_D}{\sqrt{MSE}}\right) \quad (43)$$

$$MSE = \frac{1}{M \times N \sum_{i=1}^M \sum_{j=1}^N (D(i, j) - P(i, j))^2} \quad (44)$$

Among them, $D(i, j)$ and $P(i, j)$ represent the decrypted image and the original image respectively, M and N are the length and width of the image respectively, and MAX_D is the quantized gray level of the decrypted image.

The evaluation result is expressed in dB (decibel). Generally, the greater the PSNR value between two images, the higher the degree of restoration. When the PSNR value is close to 0 dB, the image distortion is the largest. In addition to visually judging the difference between the original image and the decrypted image through the results of image simulation, this section also judges the difference between the two quantitatively. Table 9 shows the PSNR values of the original image and the decrypted image of different sizes and all black and white. Table 10 and Table 11 show the comparison with related references [38], [42], [43], [44], [45], [46].

It can be seen from Table 9 that the PSNR of the original image measured in this section and the respective decrypted

TABLE 9. MSE, PSNR table.

	Lena_256	Lena_512	Butterfly	Black	White
MSE	0	0	0	0	0
PSNR(dB)	Inf	Inf	Inf	Inf	Inf

TABLE 10. Lena_256 MSE, PSNR comparison.

	Lena_256	Ref. [44]	Ref. [45]	Ref. [46]
MSE	0	-	0	-
PSNR(dB)	Inf	13.15	6.74	22.36

TABLE 11. Lena_512 MSE, PSNR comparison.

	Lena_512	Ref. [38]	Ref. [42]	Ref. [43]
MSE	0	0	0	-
PSNR(dB)	Inf	9.81	23.87	8.94

image is relatively high, indicating that the algorithm can effectively restore the information of the plaintext image. Moreover, the comparison of Table 10 and Table 11 with the latest references can also show that decrypted image obtained by this algorithm has the lowest distortion and has a strong ability to restore.

G. DIFFERENTIAL ATTACK ANALYSIS

The meaning of the differential attack is to attack the cryptographic system by analyzing the change and propagation of the plaintext images with specific differences after being encrypted [3], [11], [35]. In order to resist the differential attack and eliminate the correlation between the plaintext image and the encrypted image, small changes in the plaintext image should cause prominent changes in the encrypted image. The pixel change rate NPCR and the uniform average change intensity UACI are usually used to test the correlation between the plaintext image and the ciphertext image. The specific definitions of the two formulas are given in section V-B. The ideal value of NPCR is 99.6054% and the ideal value of UACI is 33.4653%.

Take the Lena_512 image as a original image as an example to carry out three groups of experiments, mainly including: randomly changing one pixel value of the plaintext image to obtain the encrypted image E1, randomly changing the two pixel values of the plaintext image to obtain the encrypted image E2, and randomly changing the three pixel values of the plaintext image to obtain the encrypted image E3. Fig. 17 shows the encrypted image E generated without changing the plaintext image, the encrypted images generated by changing the original images, the difference images of the encrypted images, and the histogram of the difference images. Table 12 calculates the NPCR values and UACI values of the encrypted images E1, E2, E3 and E obtained from the three groups of experiments.

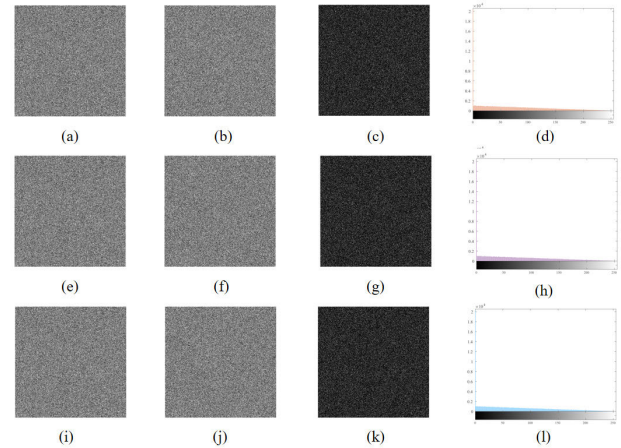


FIGURE 17. Simulation experiment of differential attack (a) encrypted image E, (b) encrypted image E1, (c) E-E1 difference image, (d) E-E1 difference image histogram, (e) encrypted image E, (f) encrypted image E2, (g) E-E2 difference image, (h) E-E2 difference image histogram, (i) encrypted image E, (g) encrypted image E3, (k) E-E3 difference image, (i) E-E3 difference image histogram.

TABLE 12. Differential attack NPCR and UACI comparison table.

Original image pixel coordinates	NPCR (%)	UACI(%)
(60,161)	99.6223	33.4659
(492,488),(118,261)	99.6123	33.4232
(10,16),(44,311),(119,393)	99.6229	33.4010

It can be seen intuitively from Fig. 17 that the encrypted image generated by the original image without changing and the encrypted image generated by the plaintext image change both show the state of snowflake noise, and there is no obvious difference visually, but through the difference images and the histogram of the difference images, it can be explained that although the ciphertext images appear to be snowflake noise images, in fact, the degree of difference between the encrypted images is very large. In addition, it can be seen from the data analysis in Table 12 that a certain or a few pixel values of the original image change, which may cause huge differences. This can further explain that subtle changes in the plaintext image will cause significant changes in the ciphertext image, and this algorithm has strong resistance to differential attacks.

H. TYPICAL ATTACK ANALYSIS

Generally speaking, only the correct key can be used to restore the ciphertext to the correct plaintext information. However, according to cryptography, an attacker can use a special technique to destroy the encryption algorithm through repeated experiments, and then obtain plaintext information. If the attacker does not know the key, he can use other effective information to attack the encryption system. Then, according to the type of information the attacker has, the attack types can be divided into four types. These four typical attack types are shown in this section [2], [7], [28]:

1) Ciphertext only attack: The attacker possesses the cryptographic algorithm and statistical characteristics of the plaintext image, and also intercepts one or more ciphertexts encrypted with the same key. The plaintext or key can be analyzed from this information. It can usually be understood as a brute force attack when only the ciphertext is known.

2) Chosen ciphertext attack: The attacker has access to the decryption system and can choose any ciphertext that is beneficial to the attack and its corresponding plaintext.

3) Known plaintext attack: The attacker obtains a certain part of the plaintext and the corresponding ciphertext, which can be any non-empty subset of the known plaintext pair, and uses this information to analyze and crack the cryptographic system.

4) Chosen plaintext attack: Chosen plaintext attack is the most typical attack method. The attacker has access to the decryption system and can select the plaintext and its corresponding ciphertext, that is, know the encrypted ciphertext and the decrypted plaintext.

Obviously, chosen plaintext attack is the strongest attack method. If the system can resist a chosen plaintext attack, it must be able to resist the other three typical attack methods. In this algorithm, the plaintext image uses the SHA-512 hash function to generate a message digest, and the message digest is used to generate the initial value of the 2D-LSM and the number of iterative discards. The chaotic system iteratively generates a chaotic sequence. There is a kind dependency of chaotic sequence and the original image. Therefore, the algorithm can resist chosen plaintext attacks, which further shows that the algorithm can also resist the other three typical attacks.

I. NOISE ATTACK ANALYSIS

In addition to having good security to resist brute force cracking by the attacker, the image encryption algorithm should also have a certain ability to resist noise attacks [29], [39]. Because the transmission channel may sometimes have a certain degree of instability, the image data may be disturbed by noise factors. The encrypted image is often very sensitive, and sometimes a little noise interference may cause the receiver to receive the corrupted encrypted image, which in turn leads to the distortion of the decrypted image. In this section, by adding speckle noise, Gaussian noise and salt and pepper noise of different intensities to the test image Lena_512 image [30], it is used to test the performance of the proposed algorithm to resist noise attacks. The experimental results are shown in Fig. 18, Fig. 19, and Fig. 20.

It can be seen from the experimental results that speckle noise has the least impact on the decrypted image, followed by salt and pepper noise, and Gaussian noise has the greatest impact on the decrypted image. As the degree of interference of salt and pepper noise increases, the affected pixels in the decrypted image gradually increase, but it does not affect the recognition of the image by the naked eye, and the rough information of the original image can still be clearly distinguished. Compared with the interference of salt and pepper

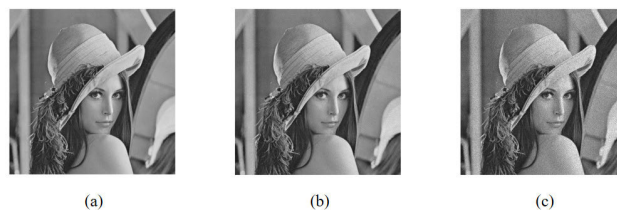


FIGURE 18. Speckle noise attack (a) 0.0001 speckle noise, (b) 0.0011 speckle noise, (c) 0.01 speckle noise.

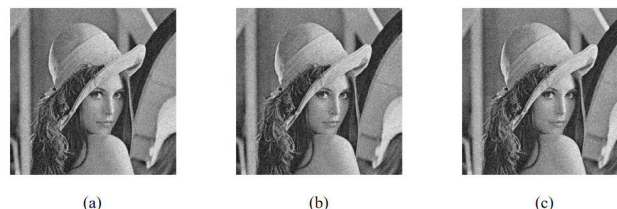


FIGURE 19. Gaussian noise attack (a) 0.0001 Gaussian noise, (b) 0.0011 Gaussian noise, (c) 0.01 Gaussian noise.

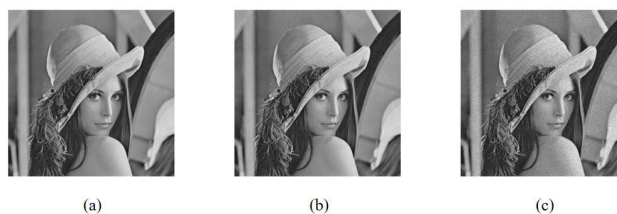


FIGURE 20. Salt and pepper noise attack (a) 0.0001 salt and pepper noise, (b) 0.0011 salt and pepper noise, (c) 0.01 salt and pepper noise.

noise, the interference of Gaussian noise has a greater impact on the decrypted image, but its impact is within an acceptable range and will not affect the recognition of the original image. Generally speaking, the three types of different levels of noise have a certain impact on the decrypted image to some extent, but the impact is very small and can be ignored. Therefore, the proposed algorithm has a strong ability to resist noise attacks.

J. CROPPING ATTACK ANALYSIS

Data loss will inevitably occur during the transmission of encrypted images, which requires the encryption and decryption schemes to be able to resist cropping attacks well. In order to prove that the proposed scheme has a certain ability to resist cropping attacks, this section tests the encrypted images and decrypted images that lose 1/16 and 1/4 of the data, as shown in Fig. 21.

It can be seen from Fig. 21 that the decrypted image can still identify the valid information of the plaintext image, indicating that the encryption algorithm can resist cropping attacks and has certain robustness.

K. RANDOMNESS ANALYSIS OF ENCRYPTED IMAGE

NIST SP800 is a series of information security guidelines issued by the National Institute of Standards and Technology

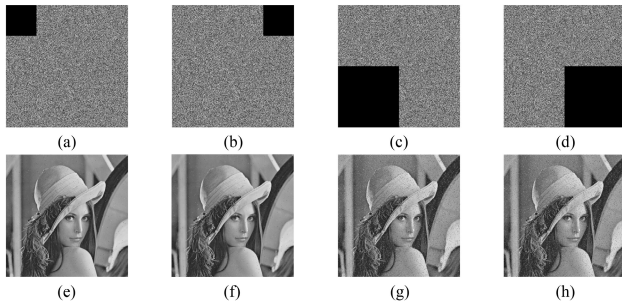


FIGURE 21. Cropping attack test results (a) 1/16 of the encrypted image is cropped in the upper left corner, (b) 1/16 of the encrypted image is cropped in the upper right corner, (c) 1/4 of the encrypted image is cropped in the lower left corner, (d) 1/4 of the encrypted image is cropped in the upper right corner, (e) decrypted image of (a), (f) decrypted image of (b), (g) decrypted image of (c), (h) decrypted image of (d).

TABLE 13. NIST test result of encrypted images.

Test index	Lena_256	Lena_512	Butterfly	Result
Frequency	0.87553	0.79230	0.85119	Pass
Block-frequency	0.39245	0.26541	0.25302	Pass
Cumulative Sums	0.63711	0.68690	0.57385	Pass
Runs	0.63711	0.64201	0.58465	Pass
Longest Run of Ones	0.63711	0.60980	0.67282	Pass
Rank	0.43727	0.46221	0.51073	Pass
FFT	0.87553	0.63967	0.78263	Pass
Non Periodic Template	0.14635	0.25023	0.19178	Pass
Overlapping Template	0.54499	0.67124	0.62304	Pass
Universal	0.07132	0.13589	0.18076	Pass
Approximate Entropy	0.01254	0.01285	0.25405	Pass
Random Excursions	0.90305	0.93684	0.98013	Pass
Excursions Variant	0.39742	0.45684	0.43443	Pass
Linear Complexity	0.19227	0.23258	0.11611	Pass
Serial	0.87483	0.88492	0.87399	Pass

(NIST), and has become a de facto standard and authoritative guide widely recognized by researchers in the information security field [13], [30]. The NIST test plan is a statistical package that includes 15 test methods. These test methods can test the randomness of arbitrary length binary sequences generated as random passwords or pseudo-random number generators based on software or hardware. These test methods are mainly used to determine the various pseudo-randomness that may exist in the sequence. This section is mainly used to test the randomness of the encrypted image, convert each pixel value of the encrypted image into 8 binary bits, and perform statistical tests on the bit sequence. Due to the number of cycles required to apply the random drift test and its variants, this test is suitable for encrypted images with a size of 512*512. The test results can be obtained from Table 13.

In Table 13, the P value of the NIST test applied to multiple encrypted images is given. Since all P values are greater than 0.01, this means that the encrypted image has passed the randomness test, and the encrypted image has high security, indicating that the proposed algorithm is safe in actual encryption applications.

TABLE 14. Time table.

Time(s)	Lena_256	Lena_512	Butterfly	Black	White
Encryption	0.0770	0.1795	0.85262	0.1901	0.1822
Decryption	0.1041	0.2356	0.9367	0.2130	0.23237
Average	0.09055	0.20755	0.89466	0.20155	0.207285

TABLE 15. Lena_256 time comparison.

Time(s)	Lena_256	Ref. [44]	Ref. [45]	Ref. [46]
Encryption	0.0770	1.9130	0.0780	0.0130
Decryption	0.1041	2.1550	0.9080	0.6232
Average	0.09055	2.0340	0.4930	0.3181

TABLE 16. Lena_512 time comparison.

Time(s)	Lena_512	Ref. [38]	Ref. [42]	Ref. [43]
Encryption	0.1795	18.9721	0.2103	0.1936
Decryption	0.2356	19.2198	0.2502	0.2283
Average	0.20755	19.09595	0.21095	0.21095

L. TIME PERFORMANCE ANALYSIS

A good encryption algorithm should have high time efficiency [38], [40]. For the proposed image encryption algorithm, under the precondition of ensuring the image transmission quality and security, if its operating efficiency is equal to or even higher than the existing encryption algorithm, the purpose of designing a new image encryption algorithm to improve transmission efficiency has been achieved.

In this section, the time performance of the encryption algorithm is analyzed. In order to verify that the proposed algorithm can evidently improve the efficiency of encryption and decryption, the encryption and decryption time of three groups of plaintext images of different sizes and all black and all white images are calculated. The test results are shown in Table 14. In addition, the comparison with related references [38], [42], [43], [44], [45], and [46] is also given, as shown in Table 15 and Table 16.

It can be seen from Table 14 that the encryption and decryption speeds of the five groups of images are all higher than the qualified minimum standard. And even if the image with the largest size is encrypted, the time required is only about 0.85262 s. This test result is very satisfactory, indicating that the proposed algorithm has passed the operational efficiency test. Moreover, no matter which type and size of the image is encrypted or decrypted, the algorithm takes a very short time, and it also meets the requirements of real-time communication of encrypted and decrypted images. In addition, the comparison in Table 15 and Table 16 also show that the efficiency of image encryption using this scheme is significantly better than that of the latest algorithm for encryption and decryption, indicating that this scheme has certain advantages in time performance.

VI. CONCLUSION

In this paper, a dual-domain image encryption algorithm based on hyperchaos and dynamic wavelet decomposition is proposed. A new encryption algorithm similar to ‘sandwich’ is realized, which combines the block scrambling in the spatial domain, the wavelet decomposition in the frequency domain, and the diffusion operation in the spatial domain. Firstly, the original image is divided into several blocks, and each matrix block is scrambled under the control of a random number sequence to obtain a scrambled matrix. Secondly, use the plaintext Hamming distance to dynamically select the wavelet type, perform wavelet decomposition on the scrambled matrix, and generate a wavelet coefficient matrix. At the same time, the key matrix is obtained by performing the improved Zigzag transformation on the scrambled matrix. Then, the initial value of the chaotic system is generated by the original image and the SHA-512 algorithm, and the chaotic key matrix is generated after the 2D-LSM iteration. Finally, the XOR operation is performed on the wavelet coefficient matrix, the chaotic key matrix, and the key matrix to obtain the encrypted image. Among them, the key is related to the plaintext image, which greatly improves the algorithm’s ability to resist known plaintext and chosen plaintext attacks. The experimental results show that the algorithm has a large key space, good encryption effect, strong randomness, and high time efficiency. In addition, correlation analysis, information entropy analysis, differential attack, and noise attack analysis are also carried out on the proposed encryption algorithm to prove that the algorithm is safe and reliable.

In future research, we will continue to study the image encryption method combining spatial domain encryption and frequency domain encryption, and strive to propose better algorithms. At the same time, we will continue to improve the robustness of the algorithm and further improve the security level of the algorithm. In addition, we will generalize the encryption algorithm to practical applications, especially to apply it to information encryption systems that require high efficiency and high security.

REFERENCES

- [1] Z. Liang, Q. Qin, and C. Zhou, “An image encryption algorithm based on Fibonacci Q-matrix and genetic algorithm,” *Neural Comput. Appl.*, vol. 34, no. 21, pp. 19313–19341, Nov. 2022, doi: [10.1007/s00521-022-07493-x](https://doi.org/10.1007/s00521-022-07493-x).
- [2] Z. Liang, Q. Qin, C. Zhou, N. Wang, Y. Xu, and W. Zhou, “Medical image encryption algorithm based on a new five-dimensional three-leaf chaotic system and genetic operation,” *PLoS ONE*, vol. 16, no. 11, Nov. 2021, Art. no. e0260014.
- [3] Z. Liang, Q. Qin, C. Zhou, and S. Xu, “Color image encryption algorithm based on four-dimensional multi-stable hyper chaotic system and DNA strand displacement,” *J. Electr. Eng. Technol.*, to be published, doi: [10.1007/s42835-022-01157-5](https://doi.org/10.1007/s42835-022-01157-5).
- [4] C. Zou, Q. Zhang, C. Zhou, and W. Cao, “A nonlinear neural network based on an analog DNA toehold mediated strand displacement reaction circuit,” *Nanoscale*, vol. 14, no. 17, pp. 6585–6599, May 2022.
- [5] X. Wang, X. Wang, B. Ma, Q. Li, and Y.-Q. Shi, “High precision error prediction algorithm based on ridge regression predictor for reversible data hiding,” *IEEE Signal Process. Lett.*, vol. 28, pp. 1125–1129, 2021.
- [6] B. Ma and Y. Q. Shi, “A reversible data hiding scheme based on code division multiplexing,” *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 1914–1927, Sep. 2016.
- [7] Q. Li, X. Wang, B. Ma, X. Wang, C. Wang, S. Gao, and Y. Shi, “Concealed attack for robust watermarking based on generative model and perceptual loss,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5695–5706, Aug. 2022.
- [8] C. Wang, B. Ma, Z. Xia, J. Li, Q. Li, and Y.-Q. Shi, “Stereoscopic image description with trinion fractional-order continuous orthogonal moments,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1998–2012, Apr. 2022.
- [9] Z. Hua, Y. Chen, H. Bao, and Y. Zhou, “Two-dimensional parametric polynomial chaotic system,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 7, pp. 6585–6599, Jul. 2021.
- [10] Y. Khedmati, R. Parvaz, and Y. Behroo, “2D hybrid chaos map for image security transform based on framelet and cellular automata,” *Inf. Sci.*, vol. 512, pp. 855–879, Feb. 2020.
- [11] C. Xiu, R. Zhou, S. Zhao, and G. Xu, “Memristive hyperchaos secure communication based on sliding mode control,” *Nonlinear Dyn.*, vol. 104, no. 1, pp. 789–805, Mar. 2021.
- [12] S. Zhu and C. Zhu, “Secure image encryption algorithm based on hyperchaos and dynamic DNA coding,” *Entropy*, vol. 22, no. 7, p. 772, Jul. 2020.
- [13] Z. Man, J. Li, X. Di, Y. Sheng, and Z. Liu, “Double image encryption algorithm based on neural network and chaos,” *Chaos, Solitons Fractals*, vol. 152, Nov. 2021, Art. no. 111318.
- [14] Z. Man, J. Li, X. Di, Y. Sheng, and Z. Liu, “Double image encryption algorithm based on neural network and chaos,” *Chaos, Solitons Fractals*, vol. 152, Nov. 2021, Art. no. 111318.
- [15] W. Feng, Y.-G. He, H.-M. Li, and C.-L. Li, “A plain-image-related chaotic image encryption algorithm based on DNA sequence operation and discrete logarithm,” *IEEE Access*, vol. 7, pp. 181589–181609, 2019.
- [16] W. Feng and Y.-G. He, “Cryptanalysis and improvement of the hyperchaotic image encryption scheme based on DNA encoding and scrambling,” *IEEE Photon. J.*, vol. 10, no. 6, pp. 1–15, Dec. 2018.
- [17] W. Feng, Y.-G. He, H.-M. Li, and C.-L. Li, “Image encryption algorithm based on discrete logarithm and memristive chaotic system,” *Eur. Phys. J. Special Topics*, vol. 228, no. 10, pp. 1951–1967, Oct. 2019.
- [18] W. Feng, Z. Qin, J. Zhang, and M. Ahmad, “Cryptanalysis and improvement of the image encryption scheme based on feistel network and dynamic DNA encoding,” *IEEE Access*, vol. 9, pp. 145459–145470, 2021.
- [19] W. Feng and J. Zhang, “Cryptanalyzing a novel hyper-chaotic image encryption scheme based on pixel-level filtering and DNA-level diffusion,” *IEEE Access*, vol. 8, pp. 209471–209482, 2020.
- [20] H. Li, T. Li, W. Feng, J. Zhang, J. Zhang, L. Gan, and C. Li, “A novel image encryption scheme based on non-adjacent parallel permutation and dynamic DNA-level two-way diffusion,” *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102844.
- [21] Y.-M. Li, D. Wei, and L. Zhang, “Double-encrypted watermarking algorithm based on cosine transform and fractional Fourier transform in invariant wavelet domain,” *Inf. Sci.*, vol. 551, pp. 205–227, Apr. 2021.
- [22] X. Yang, Q. Xue, X. Yang, H. Yin, Y. Qu, X. Li, and J. Wu, “A novel prediction model for the inbound passenger flow of urban rail transit,” *Inf. Sci.*, vol. 566, pp. 347–363, Aug. 2021.
- [23] X. Wang and S. Gao, “Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network,” *Inf. Sci.*, vol. 539, pp. 195–214, Oct. 2020.
- [24] X. Wang and X. Chen, “An image encryption algorithm based on dynamic row scrambling and zigzag transformation,” *Chaos, Solitons Fractals*, vol. 147, Jun. 2021, Art. no. 110962.
- [25] X. Wang and M. Zhang, “An image encryption algorithm based on new chaos and diffusion values of a truth table,” *Inf. Sci.*, vol. 579, pp. 128–149, Nov. 2021.
- [26] X. Wu, D. Wang, J. Kurths, and H. Kan, “A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system,” *Inf. Sci.*, vol. 349, pp. 137–153, Jul. 2016.
- [27] X. Wang, C. Liu, and D. Jiang, “A novel triple-image encryption and hiding algorithm based on chaos, compressive sensing and 3D DCT,” *Inf. Sci.*, vol. 574, pp. 505–527, Oct. 2021.
- [28] Z. Hua, K. Zhang, Y. Li, and Y. Zhou, “Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing,” *Signal Process.*, vol. 183, Jun. 2021, Art. no. 107998.
- [29] X. Yan, X. Wang, and Y. Xian, “Chaotic image encryption algorithm based on fractional order scrambling wavelet transform and 3D cyclic displacement operation,” *IEEE Access*, vol. 8, pp. 208718–208736, 2020.
- [30] H. R. Shakir, “An image encryption method based on selective AES coding of wavelet transform and chaotic pixel shuffling,” *Multimedia Tools Appl.*, vol. 78, no. 18, pp. 26073–26087, Sep. 2019.

- [31] Z. Hua, Z. Zhu, Y. Chen, and Y. Li, "Color image encryption using orthogonal Latin squares and a new 2D chaotic system," *Nonlinear Dyn.*, vol. 104, no. 4, pp. 4505–4522, Jun. 2021.
- [32] T. Yan, P. Wu, Y. Qian, Z. Hu, and F. Liu, "Multiscale fusion and aggregation PCNN for 3D shape recovery," *Inf. Sci.*, vol. 536, pp. 277–297, Oct. 2020.
- [33] D. Zhang, L. Chen, and T. Li, "Hyper-chaotic color image encryption based on transformed zigzag diffusion and RNA operation," *Entropy*, vol. 23, no. 3, p. 361, Mar. 2021.
- [34] P. Ramasamy, V. Ranganathan, S. Kadry, R. Damaševičius, and T. Blažauskas, "An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced logistic—Tent map," *Entropy*, vol. 21, no. 7, p. 656, Jul. 2019.
- [35] X. Xue, H. Jin, D. Zhou, and C. Zhou, "Medical image protection algorithm based on deoxyribonucleic acid chain of dynamic length," *Frontiers Genet.*, vol. 12, Mar. 2021, Art. no. 654663.
- [36] A. A. Rezk, A. H. Madian, A. G. Radwan, and A. M. Soliman, "On-the-fly parallel processing IP-core for image blur detection, compression, and chaotic encryption based on FPGA," *IEEE Access*, vol. 9, pp. 82726–82746, 2021.
- [37] O. P. Singh and A. K. Singh, "A robust information hiding algorithm based on lossless encryption and NSCT-HD-SVD," *Mach. Vis. Appl.*, vol. 32, no. 4, p. 101, Jul. 2021.
- [38] V. Manikandan and R. Amirtharajan, "On dual encryption with RC6 and combined logistic tent map for grayscale and DICOM," *Multimedia Tools Appl.*, pp. 23511–23540, May 2021.
- [39] K. A. Korba, D. Abed, and M. Fezari, "Securing physical layer using new chaotic parametric maps," *Multimedia Tools Appl.*, vol. 80, nos. 21–23, pp. 32595–32613, Sep. 2021.
- [40] N. Tsafack, J. Kengne, B. Abd-El-Atty, A. M. Iliyasu, K. Hirota, and A. A. A. EL-Latif, "Design and implementation of a simple dynamical 4-D chaotic circuit with applications in image encryption," *Inf. Sci.*, vol. 515, pp. 191–217, Apr. 2020.
- [41] S. Yu, N. Zhou, L. Gong, and Z. Nie, "Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyper-chaotic system," *Opt. Lasers Eng.*, vol. 124, Jan. 2020, Art. no. 105816.
- [42] T. Li, B. Du, and X. Liang, "Image encryption algorithm based on logistic and two-dimensional Lorenz," *IEEE Access*, vol. 8, pp. 13792–13805, 2020.
- [43] Y. Luo, X. Ouyang, J. Liu, L. Cao, and Y. Zou, "An image encryption scheme based on particle swarm optimization algorithm and hyperchaotic system," *Soft Comput.*, vol. 26, no. 11, pp. 5409–5435, Jan. 2022.
- [44] L. E. Reyes-López, J. S. Murguía, H. González-Aguilar, M. T. Ramírez-Torres, M. Mejía-Carlos, and J. O. Armijo-Correa, "Scaling analysis of an image encryption scheme based on chaotic dynamical systems," *Entropy*, vol. 23, no. 6, p. 672, May 2021.
- [45] M. Zhang, X. Tong, Z. Wang, and P. Chen, "Joint lossless image compression and encryption scheme based on CALIC and hyperchaotic system," *Entropy*, vol. 23, no. 8, p. 1096, Aug. 2021.
- [46] Y. Zhao and L. Liu, "A bit shift image encryption algorithm based on double chaotic systems," *Entropy*, vol. 23, no. 9, p. 1127, Aug. 2021.



ZHONGYUE LIANG was born in Binzhou, Shandong, China, in 1994. She is currently pursuing the master's degree. She has published many articles in the field of image encryption as the first author in *NCA*, *Plos one*, and other journals. Her main research interests include image encryption and DNA computing.



SHUANG LIU was born in 1977. She is an Associate Professor and the Doctor of engineering. She is currently the Deputy Dean of the School of Computer Science and Engineering, Dalian Minzu University. Her research interests include image encryption, machine learning, and video information processing.



XIAO WANG was born in Wuyi, China, in 1977. She received the master's degree in mechanical and electronic engineering from the Zhejiang University of Technology. She is a Lecturer. She has published more than ten articles. Her research interests include image encryption, intelligent computing, and computer aided design.



QIXIA QIN was born in Jinan, Shandong, China, in 1996. She is currently pursuing the master's degree. Her research interests include image encryption and artificial intelligence.



CHANGJUN ZHOU was born in 1977. He received the Ph.D. degree. He is currently a Distinguished Professor of Shuanglong Scholar with the School of Mathematics and Computer Science, Zhejiang Normal University. His research interests include image encryption, intelligent computing and pattern recognition in new computing models, biological computing theory and its application, and software system development.

...