An Image Encryption Algorithm Based on Improved Lifting-Like Structure and Cross-Plane Zigzag Transform

HAIXIAO LI\(^1\), YINAN HU\(^1\), ZIMING SHI\(^2\), BIN WANG\(^1\), AND PAN ZHENG\(^3\), (Senior Member, IEEE)

\(^1\)The Key Laboratory of Advanced Design and Intelligent Computing, School of Software Engineering, Dalian University, Dalian, Liaoning 116622, China
\(^2\)Experimental Center, Dalian University, Dalian, Liaoning 116622, China
\(^3\)Department of Accounting and Information Systems, University of Canterbury, Christchurch 8011, New Zealand

Corresponding authors: Ziming Shi (szm_zz@163.com) and Bin Wang (wangbin@dlu.edu.cn)

This work was supported in part by the National Key Technology Research and Development Program of China under Grant 2018YFC0910500; in part by the National Natural Science Foundation of China under Grant 61425002, Grant 61751203, Grant 61772100, Grant 61972266, and Grant 61802040; in part by the Liaoning Revitalization Talents Program under Grant XLYC2008017; in part by the Innovation and Entrepreneurship Team of Dalian University under Grant XQN202008; in part by the Natural Science Foundation of Liaoning Province under Grant 2021-M5-344 and Grant 2021-KF-11-03; in part by the Scientific Research Fund of Liaoning Provincial Education Department under Grant LJZKZ1186; and in part by the Dalian University Scientific Research Platform Program under Grant 202101YB02.

ABSTRACT With the increasing development of information technology, image information security has received more and more attention. Furthermore, image encryption is one of the most powerful methods to ensure image information security and plays a vital role in modern society. This paper proposes an image encryption algorithm based on an improved lifting-like structure and cross-plane zigzag transform. First, to overcome the problem of simple original lifting-like structure and single update function. We modularize the original image and add new update functions and prediction functions to handle the information between different modules. Our proposed structure has higher encryption quality and can better spread the information between each image pixel. Secondly, we propose a cross-plane zigzag transform, which can effectively eliminate the correlation between adjacent elements of the original image by applying the zigzag transform to the replacement between different planes, overcoming the disadvantage that the traditional zigzag transform is limited in scanning mode and challenging to resist brute-force attack. The experimental results show that compared with other state-of-the-art schemes, this scheme has higher information entropy, can well resist selective plaintext attacks and differential attacks, and has better security.

INDEX TERMS Image security, image encryption, chaotic system, zigzag transform.

I. INTRODUCTION

With the advancement of technology and the popularity of the Internet, people have increasingly convenient access to information. As a widely used data format, images can carry a large amount of intuitive information by itself [1]. Once this information is leaked, it is likely to cause some damage to individuals, society, and even the country. Therefore, it becomes crucial to protect the security of digital images.

To protect image information from leakage. Many new methods have emerged, such as image watermarking [2], [3], data hiding [4], [5], and image encryption has attracted extensive research as one of the most intuitive methods. Due to the large data volume, high correlation, and high redundancy of images [6], traditional data encryption algorithms such as AES, DES, RSA, and IDEA are not the optimum choices for image encryption [7]. In recent years chaotic systems have proven to be well suited for image encryption because of their extreme sensitivity to initial values, pseudo-randomness, unpredictability, and ergodicity [8]–[10]. In 1989, Matthews [11]
proposed the first encryption algorithm based on chaotic systems. Chaos-based cryptography has developed into a new branch of cryptography. In 1997 Fridrich [12] first applied the permutation diffusion structure to image encryption, and the permutation phase can well eliminate the strong correlation between adjacent pixels [13]. The diffusion phase can change the image’s pixel values and diffuse them among different pixels [14]. Subsequently, image encryption algorithms have been developed significantly [15]–[19].

Image encryption algorithms have cross-fusion gradually with other disciplines in their continuous development in recent years. More and more image encryption algorithms have been continuously proposed. Some of them are based on DNA and RNA operations [20]–[24], neural network-based [25]–[29], Latin square matrix-based [30]–[33], and compression-aware [34]–[38]. Various image encryption algorithms have emerged. A Latin square matrix-based image encryption algorithm was proposed by Xu et al. [31] in 2019, which treats the original image as a 3D matrix and permutes and diffuses it using a Latin cube, which requires the use of three Latin square matrices of size $MN$ to complete the image permutation and diffusion. Wen et al. [20] 2020 proposed a color light field image encryption scheme, which uses block processing to divide multiple sub-blocks. The data is then encrypted using DNA sequences and chaotic systems. Chaotic sequences of image size $M \times 4N$ need to be generated in the step of diffusion of image information. All the above encryption systems generate pseudo-random sequences of length no less than the normal image size in the diffusion, and since the generation and operation of many pseudo-random sequences affect the efficiency of the whole encryption system, how to use fewer pseudo-random sequences to achieve the same encryption effect has attracted extensive research. Therefore, Zhang proposed an image encryption algorithm based on a lifting scheme in 2020 [39]. The algorithm firstly decomposes the planar image into low-frequency approximate components and approximate high-frequency components, followed by the interference of the two sets of components using chaotic sequences. Finally, the encrypted image is obtained. The scheme can encrypt the original image into a meaningless random image by using a pseudo-random sequence of half the image size. Then in 2021, Zhang [40] proposed a unified image encryption system with the same encryption and decryption algorithms based on the lifting scheme. This system uses external keys and Henon mapping to generate equivalent keys. Then the image information is diffused using a lifting-like transformation. However, this system can encrypt meaningful original images into meaningless noisy images and has a strong sensitivity to both ordinary and encrypted images. However, its lifting-like structure is relatively simple, and it is easy for cryptanalysts to find the pattern which leads to the system being broken. So complex update functions and prediction functions are needed to improve the security.

“Zigzag” is often used for image encryption as a typical permutation operation [41], [42]. Wang et al. [43] 2019 proposed an image encryption algorithm based on an improved zigzag transform and a composite chaos system, which expands the original zigzag from only one scan pattern to four scan patterns. Then the pixel values are corrupted using the new chaotic encryption system and finally propagated using the adjacent pixel XOR method. In the same year, Wang et al. [44] proposed another image encryption algorithm based on the class zigzag and DNA encoding. In this algorithm, the scanning order does not start from the upper left corner of the matrix but the diagonal of the matrix. Gao et al. [45] 2021 proposed a zigzag transform with the bidirectional crossover starting from a random position. This is more complex than other zigzag transforms and can better break plaintext correlation. As improved zigzags continue to be proposed, although scanning patterns are becoming more complex, only a limited number of scanning patterns whose keyspace is not resistant to brute-force attacks [46]. Therefore, Hua et al. [46] proposed a new improved zigzag transform to solve this problem and obtain more random results, which can use pseudo-random numbers of $4 \times N$ to permute images of size $N \times N$. Compared with the traditional zigzag, the scanning order of the improved zigzag transform is controlled by parameters, which significantly improves the transform efficiency and security level. However, these methods perform the permutation in a two-dimensional plane, which results in some adjacent pixels at particular locations of the image not being permuted. Although Zhang et al. [47] applied the zigzag to a 3D cube and extended it from the 2D plane to the 3D plane, the extension was to apply the zigzag to the diffusion of the image and did not break the drawback of having only a finite scan pattern that could not resist brute-force attacks.

This paper proposes an image encryption scheme based on the improved lifting-like structure and cross-plane zigzag transform to address the shortcomings of the above image encryption algorithms. The improved lifting-like structure divides the image into four modules of the same size. It adds new update and prediction functions, which only requires $\frac{1}{4}$ chaotic sequence of image size to complete the diffusion of the original image. The encrypted image has a more significant encryption quality than the original lifting-like structure. At the same time, the cross-plane zigzag transform is applied to different planes for the first time. Firstly, we divide the original image into different modules, use the parameters generated by the chaotic system to determine each module’s scan pattern and label, and finally implement the replacement operation of different planes according to the obtained scan pattern and module label. The scheme proposed in this paper can accomplish encryption using only pseudo-random sequences one quarter of the image size. And compared with other excellent schemes, our proposed scheme has better information entropy and more significant average neighborhood grayscale difference, can well resist selective plaintext attack and differential attack, and has better security, can be applied to actual image encryption.
The rest of this paper is as follows. Section II describes the chaotic system, lifting-like structure, and cross-plane zigzag scanning. Section III describes the proposed encryption scheme. Section IV provides experimental results and detailed performance analysis. Section V draws the conclusions.

II. THEORETICAL THEORY

This section describes the four-dimensional hyperchaotic Chen system used, the lifting-like structure, and the cross-plane zigzag transform. The lifting-like structure first divides the image into four modules and then diffuses to change the image’s pixel values. The cross-plane zigzag transform can change the pixel values efficiently.

A. 4D HYPERCHAOTIC CHEN SYSTEM

Image encryption methods based on chaotic systems have become one of the most popular methods due to their high sensitivity to initial conditions, ergodicity, and unpredictability. Many researchers have proposed many encryption schemes based on chaotic systems. Compared with the low-dimensional chaotic system, the high-dimensional chaotic system has more complex chaotic behavior and more system parameters, so it has a larger key space and therefore it has higher security. Therefore, the four-dimensional hyperchaotic Chen system [48] is chosen as the chaotic vector in this paper. It is shown below:

$$\begin{align*}
\frac{dx_1}{dt} &= mx_2 - x_1 + x_4 \\
\frac{dx_2}{dt} &= -x_1 x_3 + q x_1 + p x_2 \\
\frac{dx_3}{dt} &= x_1 x_2 - n x_3 \\
\frac{dx_4}{dt} &= x_2 x_3 + r x_4
\end{align*}$$

(1)

In equation (1), $\frac{dx_i}{dt}$, $i = 1, 2, 3, 4$ is the derivative for time $t$ and $m, n, p, q, r$ are the chaotic system’s parameters, which has two positive Lyapunov exponents when $m = 35, n = 3, p = 12, q = 7, r = 0.3$, which implies that the system is hyperchaotic [49].

B. IMPROVED LIFTING-LIKE TRANSFORMATION STRUCTURE

In recent years, how to find an efficient and secure image encryption system has attracted extensive research. Zhang [40] proposed a structure for lifting-like. The structure possesses two modules, namely, a positive transform module and an inverse transform module, and the operations of the two modules are inverse. Two simple linear functions internally operate the modules for diffusion. Although the structure can change the original image pixel values with only a chaotic sequence half of the original image size, the original lifting-like transform is easy for cryptanalysts to discover the pattern and thus lead to the system being cracked due to its simple structure, so complex update functions are needed to improve the security. To solve this problem, this scheme creates two new update prediction functions while enhancing the lifting-like structure to compensate for the shortcomings of the original system, which improves the encryption efficiency while exhibiting better security. Figure 1 shows the structure of the improved lifting-like transformation.

In Figure 1, the Type-I part is the positive conversion module, and the Type-II part is the inverse conversion module. Where $+, -$ is the addition and subtraction operation of mode 256. $\{p_i\}, i = 1, 2, 3, \ldots, MN$, and $\{x_j\}, j = 1, 2, 3, \ldots, MN/4$ are the input sequences and $\{y_i\}, i = 1, 2, 3, \ldots, MN$ is the output sequence, and the detailed implementation is described next.

(i) Positive transformation module (Type-I)

The steps of the positive transformation module are as follows. Where $\{a_j\}, \{b_j\}, \{c_j\}, \{d_j\}$ are the original sequence and $\{\tilde{a}_j\}, \{\tilde{b}_j\}, \{\tilde{c}_j\}, \{\tilde{d}_j\}$ are the encrypted sequence. When subscript $j = 0$ or $j = MN/4 + 1$, we take the sequence value as zero.

Step 1. Divide the sequence $\{p_i\}, i = 1, 2, 3, \ldots, MN$ into four sequences of equal length $\{a_j\}, \{b_j\}, \{c_j\}, \{d_j\}, j = 1, 2, 3, \ldots, MN/4$.

Step 2. Obtain $\{\hat{a}_j\}$ by $\{a_j\}$ and $\{x_j\}$.

$$\hat{a}_j = a_j + x_j$$

(2)

Step 3. Obtain $\{\tilde{b}_j\}$ by $\{b_j\}$ and $\{\hat{a}_j\}$.

$$\tilde{b}_j = b_{j-1} + b_j + P(\hat{a}_j) = b_{j-1} + b_j + (\hat{a}_j + a_{j+1})/2$$

(3)

Step 4. Obtain $\{\hat{c}_j\}$ by $\{c_j\}$ and $\{\tilde{b}_j\}$.

$$\hat{c}_j = c_j + Q(\tilde{b}_j) = c_j + b_{j-1} b_{j+1} 256$$

(4)

Step 5. Obtain $\{\tilde{a}_j\}$ by $\{\hat{a}_j\}$ and $\{\tilde{b}_j\}$.

$$\tilde{a}_j = \hat{a}_j + \hat{a}_{j-1} + U(\tilde{b}_j) = \hat{a}_j + \hat{a}_{j-1} + (b_{j-1} + \tilde{b}_j)/4$$

(5)

Step 6. Obtain $\{\tilde{d}_j\}$ by $\{\hat{c}_j\}$ and $\{d_j\}$.

$$\tilde{d}_j = d_{j-1} + d_j + Q(\tilde{c}_j) = d_{j-1} + d_j + \text{mod}(\hat{c}_j \times c_{j+1}, 256)$$

(6)

Step 7. Obtain $\{\tilde{c}_j\}$ by $\{\tilde{c}_j\}$ and $\{\tilde{d}_j\}$.

$$\tilde{c}_j = \hat{c}_j + c_{j-1} + R(d_{j})$$

$$= \hat{c}_j + c_{j-1} + \text{mod}(d_{j-1} + d_j, 256)$$

(7)

$\{\tilde{a}_j\}, \{\tilde{b}_j\}, \{\tilde{c}_j\}, \{\tilde{d}_j\}, j = 1, 2, 3, \ldots, MN/4$ is the final sequence obtained by the positive transformation module.

(ii) Inverter module (Type-II)

The steps of the inverter transformation module are as follows. Where $\{\tilde{a}_j\}, \{\tilde{b}_j\}, \{\tilde{c}_j\}, \{\tilde{d}_j\}$ are the encrypted sequence
and \( \{a_j\}, \{b_j\}, \{c_j\}, \{d_j\} \) are the decryption sequence. When subscript \( j = 0 \) or \( j = MN/4 + 1 \), we take the sequence value as zero.

Step 1. Obtain \( \{\hat{c}_j\} \) by \( \{d_j\} \) and \( \{\hat{c}_j\} \).
\[
\hat{c}_j = c_j - c_j^{\hat{\ }} - R(d_j) = c_j - c_j^{\hat{\ }} - \text{mod}(d_j + d_{j-1}, 256)
\]  

(8)

Step 2. Obtain \( \{d_j\} \) by \( \{d_j\} \) and \( \{\hat{c}_j\} \).
\[
d_j = d_j - d_{j-1} - Q(c_j) = d_j - d_{j-1} - \text{mod}(c_j \times c_j^{\hat{\ }}, 256)
\]  

(9)

Step 3. Obtain \( \{\hat{a}_j\} \) by \( \{\tilde{a}_j\} \) and \( \{\tilde{b}_j\} \).
\[
\hat{a}_j = \tilde{a}_j - \tilde{a}_j^{\hat{\ }} - U(b_j) = \tilde{a}_j - \tilde{a}_j^{\hat{\ }} - (b_j - b_{j-1} + b_j)/4
\]  

(10)

Step 4. Obtain \( \{c_j\} \) by \( \{\hat{c}_j\} \) and \( \{\hat{b}_j\} \).
\[
c_j = \hat{c}_j - Q(b_j) = \hat{c}_j - \text{mod}(b_j \times b_j^{\hat{\ }}, 256)
\]  

(11)

Step 5. Obtain \( \{b_j\} \) by \( \{b_j\} \) and \( \{\hat{a}_j\} \).
\[
b_j = \tilde{b}_j - b_{j-1} - P(\hat{a}_j) = \tilde{b}_j - b_{j-1} - (\hat{a}_j + \hat{a}_j^{\hat{\ }})/2
\]  

(12)

Step 6. Obtain \( \{a_j\} \) by \( \{\hat{a}_j\} \) and \( \{x_j\} \).
\[
a_j = \hat{a}_j - x_j
\]  

(13)

Step 7. Combine the four obtained sequences \( \{a_j\}, \{b_j\}, \{c_j\}, \{d_j\} \) into sequence \( \{p_i\}, i = 1, 2, 3, \ldots, MN \).

The above procedure introduces the positive and inverse transform modules in the lifting-like structure. It can be observed that the two modules are opposite each other, and the sequence encrypted by the positive transform module can be recovered losslessly by the inverse transform module. When encrypting the same image using only the positive transform module, the improved structure, as shown in Table 1, can result in a more significant encryption quality (EQ) of the encrypted image [50]. The encryption quality can be determined by Equation (14).
\[
EQ = \sum_{L=0}^{255} \frac{(H_L(E) - H_L(I))^2}{256}
\]  

(14)

where \( E(i, j) \) and \( I(i, j) \) are the pixel gray level values at \((i, j)\) for the encrypted image and the original image for \(M \times N\) pixel \(L\) gray level, respectively. Define \( H_L(I) \) and \( H_L(E) \) as the number of occurrences of each gray level \(L\) in the encrypted image and the original image. The larger the value of EQ, the better the security of the encryption.
C. CROSS-PLANE ZIGZAG TRANSFORMATION

The zigzag transform is a process that scans the elements of a matrix in a "Z" shape and stores the order of the scanned elements in a one-dimensional array, which can then be rearranged into a two-dimensional matrix according to specific requirements [43]. Figure 2 illustrates the process of a standard zigzag transformation [43]. In general, there are eight zigzag transformation modes. As shown in Figure 3 [46], the scan can be done from four corners in horizontal or vertical directions, and the one-dimensional array can be shifted to improve the randomness.

However, the above zigzag transform still has some problems; the transform can only be performed in the same plane but not across planes, the position of some elements does not change after the zigzag transform is completed, the original zigzag transform does not break the correlation between adjacent pixels well as shown in Figure 4, and since the original zigzag transform has only a limited number of scan patterns, it is not enough to resist brute-force attacks [46]. We propose a cross-plane zigzag transform to solve these problems and obtain more scanning results. The security level is greatly improved by introducing parameters to control the scanning order. Compared with the traditional zigzag transform, extending the two-dimensional zigzag transform to three-dimensional multiplanar dislocation dramatically increases the available scanning patterns and can be resistant to brute-force attacks. The detailed steps of the cross-plane zigzag transformation are shown below.

Step 1: The image is divided into four modules upper-left (LU), upper-right (RU), down-left (LD), and down-right (RD). Generate a one-dimensional chaotic sequence \(k(i = 1, 2, 3, \ldots, 16)\) and four chaotic sequences \(A, B, C, D\) of length \(MN/4\), rang \([1, 4]\), and convert them into two-dimensional matrices of size \(\frac{M}{2} \times \frac{N}{2}\).

Step 2: Obtain \(a_j, b_j, c_j, d_j, j = 1, 2, 3, 4\) from Equation (15), where \(a_j, b_j, j = 1, 2, 3, 4\) determines the scan mode of each module separately according to Table 2. \((c_j, d_j), j = 1, 2, 3, 4\) determines the labeling of each module according to Table 3 (labeling in the order of LU, RU, LD, RD).

\[
\begin{align*}
    a_j &= (k_i \mod 2) + 1, \quad i = 1, 2, 3, 4 \\
    b_j &= (k_i \mod 2) + 1, \quad i = 5, 6, 7, 8 \\
    c_j &= (k_i \mod 4) + 1, \quad i = 9, 10, 11, 12 \\
    d_j &= (k_i \mod 6) + 1, \quad i = 13, 14, 15, 16
\end{align*}
\] (15)

Step 3: Perform a cross-plane zigzag transformation of the LU module with matrix A according to the determined scan order \((a_1, b_1)\) and label order \((c_1, d_1)\). If the first number in matrix A is 2, the first-pixel value of the LU module scan mode is exchanged with the first-pixel value of the scan mode of the module labeled 2. If the second number in matrix A is 3, the second-pixel value of the LU module scan mode is exchanged with the second-pixel value of the module scan mode labeled 3. And so on, until the LU module is finished scanning.

Step 4: Perform the cross-plane transformation of the RU module with matrix B according to the determined scan order \((a_2, b_2)\) and label order \((c_2, d_2)\).

Step 5: Perform the cross-plane transformation of the LD module with matrix C in the determined scan order \((a_3, b_3)\) and label order \((c_3, d_3)\).

Step 6: Perform the cross-plane transformation of the RD module with matrix D according to the determined scan order \((a_4, b_4)\) and label order \((c_4, d_4)\).

To recover the scrambled image, all you need to do is to use the cross-plane zigzag inversion, that is, to apply cross-plane zigzag scanning to the four modules RD, LD, RU, and LU in order according to the defined scanning mode and scanning order. To better explain the cross-plane zigzag transformation, an example image of size \(10 \times 10\) is provided. According to \(a_j = (3, 2, 2, 4), b_j = (1, 1, 2, 2), j = 1, 2, 3\) and \(c_j = (2, 3, 4, 1), d_j = (1, 6, 2, 5), j = 1, 2, 3\), each module’s scanning pattern and numbering order are determined. The permutation results of Fig. 5 are then obtained as described above.

Compared with the existing zigzag transform, the cross-plane zigzag transform not only changes the position of all pixel values but also can obtain a different scanning order by changing the initial state of the chaotic sequence, which greatly increases the keyspace and can resist brute-force attacks and increases the difficulty for attackers to find out the pattern of the cross-plane transform. Therefore, our proposed scheme has higher security.

III. AN IMAGE ENCRYPTION ALGORITHM BASED ON IMPROVED LIFTING-LIKE STRUCTURE AND CROSS-PLANE ZIGZAG TRANSFORM

This section introduces a new image encryption algorithm based on an improved lifting-like structure and cross-plane
zigzag transform, as shown in Figure 6, which presents the entire structure of the image encryption algorithm. Where +, − are the addition and subtraction operation of mode 256. \( P \) is the original image, and \{\( x_j \), \( j = 1, 2, 3, \ldots, MN/4 \} \) are the input sequences and \( Y \) is the encrypted image, and the detailed implementation is described next.

The algorithm mainly includes pseudo-random sequence generation, positive lifting-like transformation, cross-plane zigzag transformation, and inverse lifting-like transformation. This image encryption algorithm can encrypt the original image \( P \) of size \( M \times N \) into an encrypted image \( Y \) that can resist various attacks. Next, we will describe each encryption step in detail.

\[ K_3 = \{k_{67}k_{66}k_{65} \ldots \ldots k_{256}\}, \quad K_4 = \{k_{257}k_{258}k_{259} \ldots \ldots k_{128}\} \]

\[ x_0 = \text{mod}(\text{mean}(K_1), 1) \]
\[ y_0 = \text{mod}(\text{mean}(K_2), 1) \]
\[ z_0 = \text{mod}(\text{mean}(K_3), 1) \]
\[ h_0 = \text{mod}(\text{mean}(K_4), 1) \]

Step 2: The control parameters are brought into the hyperchaotic system to generate four chaotic sequences \[ \{\text{Seq}_1, \text{Seq}_2, \text{Seq}_3, \text{Seq}_4\} \] of length \( MN/16 + 4 \), which are transformed according to Equation (17) to obtain the final pseudo-random sequence.

\[ \text{Seq}_1(i) = \text{mod}(\text{abs}(\text{floor}([\text{Seq}_1(i) \times 10^{10}])), 256) \]
\[ \text{Seq}_2(i) = \text{mod}(\text{abs}(\text{floor}([\text{Seq}_2(i) \times 10^{10}])), 256) \]
\[ \text{Seq}_3(i) = \text{mod}(\text{abs}(\text{floor}([\text{Seq}_3(i) \times 10^{10}])), 256) \]
\[ \text{Seq}_4(i) = \text{mod}(\text{abs}(\text{floor}([\text{Seq}_4(i) \times 10^{10}])), 256) \]

Step 3: The last four values of \[ \{\text{Seq}_1, \text{Seq}_2, \text{Seq}_3, \text{Seq}_4\} \] are taken separately to form a new sequence \[ \{a_i, b_j, c_j, d_i\} \], and then the final zigzag transformation control parameters are
B. POSITIVE LIFTING-LIKE TRANSFORMATION

The lifting-like positive transformation can well diffuse the information of the pseudo-random sequence into the whole image, leading to change the pixel value of the whole image. Compared with the traditional diffusion method, we only need to use one-fourth of the image size of the pseudo-random sequence to complete the diffusion of the whole image, which greatly improves the efficiency of the algorithm. The detailed steps of the class boosting positive transformation are as follows.

Step 1: The input image is divided into four modules, that is, upper-left (LU) module, upper-right (RU) module, down-left (LD) module, down-right (RD) module, and the four modules are transformed into a one-dimensional sequence $\{lu_i\}_{i=1}^{MN}$, $\{ru_i\}_{i=1}^{MN}$, $\{ld_i\}_{i=1}^{MN}$, $\{rd_i\}_{i=1}^{MN}$.

Step 2: Take the first $MN/16$ values of $\{Seq_1, Seq_2, Seq_3, Seq_4\}$ respectively and form a new sequence $\{x_i\}_{i=1}^{MN/4}$.

Step 3: The resulting final sequence $\{x_i\}$ and the divided four one-dimensional sequences $\{lu_i\}_{i=1}^{\text{ru}_i}, \{ld_i\}_{i=1}^{\text{rd}_i}$ are brought into the lifting-like transform module for encryption, to obtain the final four encrypted sequences $\{lu_i'\}_{i=1}^{\text{ru}_i}, \{ld_i'\}_{i=1}^{\text{rd}_i}$.

C. CROSS-PLANE ZIGZAG TRANSFORMATION

The cross-plane zigzag transform is the first time applying the zigzag transform between different planes, which can better reduce the correlation between elements in different planes. The permutation process is controlled by chaotic sequences and scanning patterns, which improves the security level of the encryption structure. To better resist the selective plaintext attack, we divide the overall image into two planes $P_1, P_2, P_1$ containing the decimal information of the original image, and $P_2$ containing the single-digit information of the original image. The mean value of $P_1, P_2$ is used as part of the initial state of the chaotic mapping. Since the mean values of the two planes do not change during the permutation process, the forward process has the same key as the reverse process. The detailed steps of the cross-plane zigzag are as follows.

Step 1: Convert the resulting new encrypted sequence $\{lu_i'\}_{i=1}^{\text{ru}_i}, \{ld_i'\}_{i=1}^{\text{rd}_i}$ into a two-dimensional matrix of size $M_1 \times N_1$, respectively.

Step 2: Combine the four two-dimensional matrices in the order of first and last to become a new two-dimensional matrix $P' = [lu_i', ru_i', ld_i', rd_i']$, with the size $M_1 \times N_1$.

Step 3: Obtain the decimal plane $P_1$ and the single-digit plane $P_2$ of the original image, get the total value of pixels $x_1, x_2$ in both planes, and convert them to decimals according to Equation (19). Since the $P_1, P_2$ plane contains all the information of the image, a very slight change in the image will cause a change in the initial value of the hyperchaotic mapping, which generates an entirely different chaotic sequence, and thus can resist well the selection of explicit attacks.

\begin{align*}
  x_1 &= \text{mod}\left(\frac{x_1 + x_0}{2}, 1\right) \\
  x_2 &= \text{mod}\left(\frac{x_1 + x_2}{2}, 1\right)
\end{align*}

(19)

Step 4: Obtain the new chaotic system initial value $x_1, y_1, z_1, h_1$ according to Equation (20) and bring it into the chaotic system to generate four one-dimensional chaotic sequences $q_1, q_2, q_3, q_4$ of length $MN/4$.

\begin{align*}
  x_1 &= \text{mod}\left(\frac{x_1 + x_0}{2}, 1\right) \\
  y_1 &= \text{mod}\left(\frac{x_1 + y_0 + x_2}{3}, 1\right) \\
  z_1 &= \text{mod}\left(\frac{x_1 + y_1}{1}, 1\right) \\
  h_1 &= \text{mod}\left(\frac{x_1 + y_1 + z_1}{1}, 1\right)
\end{align*}

(20)

Step 5: Transform the desired chaotic sequence into the sequence required for the cross-plane zigzag transformation according to Equation (21), and then transform all the sequences into a two-dimensional matrix of $M_2 \times N_2$.

\begin{align*}
  q_1 &= \text{mod}\left(\text{abs(floor}(q_1 \times 10^{10})), 4\right) + 1 \\
  q_2 &= \text{mod}\left(\text{abs(floor}(q_2 \times 10^{10})), 4\right) + 1 \\
  q_3 &= \text{mod}\left(\text{abs(floor}(q_3 \times 10^{10})), 4\right) + 1 \\
  q_4 &= \text{mod}\left(\text{abs(floor}(q_4 \times 10^{10})), 4\right) + 1
\end{align*}

(21)

Step 6: The resulting matrix $q_1, q_2, q_3, q_4$ is placed in the cross-plane zigzag module for encryption, together with the matrix $lu_i', ru_i', ld_i', rd_i'$ generated by the class lifting positive transformation. The encryption operations performed in this module are the cross-plane zigzag transform, the flip operation for each row of each module, and the cross-plane zigzag inverse transform.

The final output is four 2D matrices $LU, RU, LD, RD$ of size $M_2 \times N_2$. Since image information is used as part of the initial value of the chaos, a tiny change in the image will cause a significant change in the final 2D matrix. Therefore, this scheme is resistant to selective plaintext attacks and has good security.

D. INVERSE LIFTING-LIKE TRANSFORMATION

First, all $LU, RU, LD, RD$ is converted into a one-dimensional matrix and then put into the inverse lifting-like transformation module with sequence $\{x_i\}$ for encryption to generate four one-dimensional sequences $LU', RU', LD', RD'$. All of them are converted into a two-dimensional matrix of size $M_4 \times N_4$, and the four matrices are combined into a new two-dimensional matrix $Y$ of size $M \times N$ in the order of upper-left, upper-right, down-left, and down-right. The resulting matrix $Y$ is the final encrypted image.
FIGURE 7. Simulation results of the proposed image encryption algorithm: (a) is the original image, (b) is the encrypted image, (c) is the decrypted image (d) is the difference image between the original image and the decrypted image |a − c|.

FIGURE 8. Histogram analysis: (a-e) histogram of the original image, (f-j) histogram of the encrypted image.

IV. EXPERIMENT AND ALGORITHM PERFORMANCE ANALYSIS

A. IMAGE ENCRYPTION AND DECRYPTION TEST

This computer is equipped with Intel(R) Core (TM) i5-9400CPU@2.90GHz 64-bit operating system, x64-based processor, and the programming language used is MATLAB. The steps used for the simulation tests are described in Section 3 above. We experiment with the system proposed in this paper with some commonly used test images, encrypting and decrypting the original images using the cryptosystem proposed in this paper. (As shown in Figure 7)

It can be seen from the above figure that the algorithm encrypts the original image as an encrypted image without any helpful information, and it can be seen from Figure VII(d)
that the difference image between the original image and the encrypted image is a pure black image so that this scheme can recover the image losslessly. It shows that this system can be fully applied to image encryption.

B. EFFICIENCY ANALYSIS
The time complexity analysis is related to the number of steps and operations needed in encryption and decryption. In the phase of generating pseudo-random sequences, the time complexity is \( O(\frac{1}{4}MN) \). The time complexity in the diffusion phase is \( O(MN) \), and the time complexity in the permutation phase is \( O(MN \log MN) \). Thus, our total time complexity is \( O(\frac{1}{4}MN + MN + MN \log MN) \). In the following Table 4, we list the comparison of the time complexity with other good algorithms and our proposed scheme has the lowest time complexity.

C. HISTOGRAM ANALYSIS
The histogram can visually display the statistical information of the image, reflecting the distribution of each gray value in the image. The histogram of the original image has apparent fluctuations, so it will show obvious statistical patterns, so it is easy for attackers to extract the transformation relationship between the plaintext image and the ciphertext image from it and use statistical attacks to break the image. To resist the statistics more effectively, the pixel histogram of the encrypted image must be homogeneous. The more uniform the histogram distribution is, the less statistical information the image shows, and the more excellent the resistance to statistical attacks. From Figure 8, we can see that the histogram of the encrypted image of this scheme is very smooth and can resist the statistical analysis attack very well.

D. INFORMATION ENTROPY ANALYSIS
Entropy is often used to describe the complexity of things, and information entropy is a quantitative measure of the degree of randomness of a signal source. That is, information entropy can be used to measure the randomness of an image. For an 8-bit grayscale image, the value of the maximum information entropy can be used to measure the randomness of an image. The grayscale difference \([58]\) is another statistical measure to a large amount of redundancy in the image, the information entropy is less than 8. The information entropy is calculated as shown in equation (22).

\[
H = - \sum_{i=0}^{255} p_i \log_2 p_i \tag{22}
\]

\(p_i\) is the probability of occurrence of a pixel with gray value \(i\).

Table 5 shows the information entropy values of our proposed image encryption scheme and compares the information entropy of different good algorithms based on the same dataset USCSIP “Miscellaneous”. From Table 5, we can see that our proposed scheme’s average value of information entropy is closer to 8. This indicates that our algorithm performs better than other algorithms and achieves better security.

| Algorithms            | Time complexity                  |
|-----------------------|----------------------------------|
| Proposed              | \(O(\frac{1}{4}MN + MN + MN \log MN)\) |
| Ref. [51]             | \(O(BMN)\)                       |
| Ref. [52]             | \(O(4(M \log N + M + N))\)       |
| Ref. [53]             | \(O(M \log 8N + 8N \log M + M + 8N)\) |

E. AVERAGE NEIGHBORHOOD GRAY LEVEL DIFFERENCE (GVD)
The grayscale difference \([58]\) is another statistical measure for comparing the randomness of the original and encrypted images and can be expressed by Equation (23).

\[
GN(x, y) = \sum \frac{(G(x, y) - G(x', y'))^2}{4} \text{ here } (x', y')
= \begin{cases} 
(x - 1, y) \\
(x + 1, y) \\
(x, y + 1) \\
(x, y - 1) 
\end{cases} \tag{23}
\]

where \(G(x, y)\) denotes the gray value of the pixel at position \((x, y)\). The average neighborhood gray difference (GVD) of the whole image \([46]\) can be calculated by Equation (24).

\[
GVD = \frac{AN'[GN(x, y)] - AN[GN(x, y)]}{AN'[GN(x, y)] + AN[GN(x, y)]} \tag{24}
\]

In equation (24), \(AN\) and \(AN'\) represent the average neighborhood gray value, the former represents the average neighborhood gray value before encryption, and the latter represents the average neighborhood gray value after encryption. The result of the above is the GVD score. If the two images are identical, then it is 0. Otherwise, it is 1. From Table 6, we can see that the GVD score of our proposed algorithm is closer to 1 compared to the GVD scores of other excellent algorithms, which indicates that our scheme can encrypt the original image well into a completely uncorrelated and meaningless image.

F. ADJACENT PIXEL CORRELATION ANALYSIS
Adjoint pixel correlation can intuitively reflect the difference between the original and encrypted images. Since there is much redundant information in the original image, the adjacent pixels will strongly correlate. When the original image is encrypted, the redundant information between neighboring pixels is reduced, so the correlation between neighboring pixels of the encrypted image is small. The correlation coefficient of adjacent pixels is calculated by the formula (25), as shown at the bottom of the next page.

\[
\text{cov}(u, v) = \text{covariance coefficient of vector } u \text{ and vector } v, u_i, v_i \text{ is each component of vector } u, v, n \text{ is the length of vector } u, v, \text{ and } E(u), E(v) \text{ is the mean of all components of each of vector } u, v. \text{ The value of } r_{uv} \text{ is between } -1 \text{ and } 1, \text{ and the closer the absolute value } |r_{uv}| \text{ is to } 1 \text{ indicates a
greater correlation of vector \( u, v \). The closer to zero means that the correlation is smaller. We take the Lena image as an example, and the results are shown in Table 7. We can see that the correlation coefficient of the original image is close to 1 in all directions, and the correlation coefficient of the encrypted image is almost zero. By comparing with other schemes, our proposed scheme can also break the correlation between neighboring pixels.


given

\[
E(x) = \frac{1}{n} \sum_{i=1}^{n} x_i
\]

\[
\text{cov}(u, v) = \frac{\sum_{i=1}^{n} (u_i - E(u))(v_i - E(v))}{\sqrt{\sum_{i=1}^{n} (u_i - E(u))^2} \sqrt{\sum_{i=1}^{n} (v_i - E(v))^2}}
\]

(25)

TABLE 5. Information entropy of different encryption algorithms.

| Test image | Propos ed | Ref. [54] | Ref. [55] | Ref. [19] | Ref. [56] | Ref. [57] |
|------------|-----------|-----------|-----------|-----------|-----------|-----------|
| 5.1.09     | 7.9974    | 7.9966    | 7.9970    | 7.9974    | 7.9971    | 7.9965    |
| 5.1.10     | 7.9975    | 7.9971    | 7.9969    | 7.9975    | 7.9971    | 7.9972    |
| 5.1.11     | 7.9973    | 7.9975    | 7.9974    | 7.9971    | 7.9973    | 7.9973    |
| 5.1.12     | 7.9974    | 7.9972    | 7.9970    | 7.9975    | 7.9968    | 7.9954    |
| 5.1.13     | 7.9974    | 7.9965    | 7.9974    | 7.9974    | 7.9974    | 7.9965    |
| 5.1.14     | 7.9974    | 7.9977    | 7.9969    | 7.9975    | 7.9970    | 7.9973    |
| 5.2.08     | 7.9993    | 7.9991    | 7.9993    | 7.9993    | 7.9993    | 7.9991    |
| 5.2.09     | 7.9993    | 7.9992    | 7.9993    | 7.9992    | 7.9993    | 7.9990    |
| 5.2.10     | 7.9993    | 7.9993    | 7.9993    | 7.9993    | 7.9993    | 7.9993    |
| 5.3.01     | 7.9998    | 7.9998    | 7.9998    | 7.9998    | 7.9998    | 7.9998    |
| 5.3.02     | 7.9998    | 7.9996    | 7.9998    | 7.9998    | 7.9998    | 7.9992    |
| 7.1.01     | 7.9993    | 7.9990    | 7.9994    | 7.9992    | 7.9992    | 7.9980    |
| 7.1.02     | 7.9993    | 7.9991    | 7.9994    | 7.9993    | 7.9993    | 7.9994    |
| 7.1.03     | 7.9994    | 7.9990    | 7.9994    | 7.9994    | 7.9993    | 7.9983    |
| 7.1.04     | 7.9993    | 7.9992    | 7.9992    | 7.9992    | 7.9992    | 7.9895    |
| 7.1.05     | 7.9994    | 7.9992    | 7.9993    | 7.9992    | 7.9994    | 7.9988    |
| 7.1.06     | 7.9993    | 7.9992    | 7.9994    | 7.9993    | 7.9994    | 7.9990    |
| 7.1.07     | 7.9994    | 7.9991    | 7.9993    | 7.9993    | 7.9993    | 7.9987    |
| 7.1.08     | 7.9994    | 7.9990    | 7.9993    | 7.9993    | 7.9993    | 7.9988    |
| 7.1.09     | 7.9994    | 7.9991    | 7.9993    | 7.9993    | 7.9993    | 7.9988    |
| 7.1.10     | 7.9993    | 7.9990    | 7.9992    | 7.9992    | 7.9993    | 7.9998    |
| 7.2.01     | 7.9998    | 7.9996    | 7.9998    | 7.9998    | 7.9998    | 7.9988    |
| Boat.512   | 7.9994    | 7.9992    | 7.9993    | 7.9992    | 7.9992    | 7.9993    |
| Gray21.5   | 7.9993    | 7.9993    | 7.9993    | 7.9993    | 7.9993    | 7.9993    |
| Ruler.51   | 7.9993    | 7.9987    | 7.9993    | 7.9993    | 7.9993    | 7.9978    |
| Mean       | 7.9989    | 7.9986    | 7.9988    | 7.9989    | 7.9988    | 7.9977    |

TABLE 6. GVD fraction.

| Images     | GVD proposed | Ref. [59] | Ref. [60] | Ref. [61] |
|------------|--------------|-----------|-----------|-----------|
| Lena512    | 0.98194      | 0.96619   | 0.96688   | 0.96660   |
| Baboon512  | 0.95260      | 0.92200   | 0.92530   | 0.92510   |
| Cameraman512 | 0.96161   | 0.92581   | 0.92590   | 0.92570   |
| Peppers512 | 0.97858      | 0.96777   | 0.96750   | 0.96720   |

To analyze whether our proposed scheme can effectively reduce the correlation between adjacent pixels, we first plotted the distribution of adjacent pixel pairs in the normal and encrypted images. The pixel pairs of the original image are mainly distributed on the diagonal, as shown in Figure 9, which indicates the strong correlation between the adjacent pixels of the original image. Then the pixel pairs of the encrypted image are randomly distributed in the whole plane. This indicates that the neighboring pixels in the encrypted image are uncorrelated.

G. KEY SENSITIVITY

A good encryption algorithm should have a strong sensitivity to the key. Otherwise, its actual key space may be much smaller than the theoretical key space. This increases the possibility of being compromised by brute force attacks.

Suppose the algorithm is sensitive to the key. A small change in the key can result in a completely different encrypted image, and a completely different image can be decrypted during the decryption process. Therefore, we design several experiments to test whether the image encryption algorithm proposed in this paper is highly sensitive to the key. First, we randomly generate a key \( K_1 \), and then obtain the other two keys, \( K_2 \) and \( K_3 \), by changing one of the bits in \( K_1 \). The obtained keys are denoted as

\[
K_1 = 3f5bace3dd6db619c49e8f2b17bce84250
\times cc0dde10bececd4bbca405f1c5cc
\times 1960f675fc89949e47c59eef90a0846
\times ed37ec4d2948981e083e32fc457c836

82314

VOLUME 10, 2022
FIGURE 9. Distribution of adjacent pixel values: a-c original image vertical, horizontal and diagonal directions, d-f encrypted image vertical, horizontal and diagonal directions.

FIGURE 10. Key sensitivity analysis in the encryption process. (a) plain image $P$; (b) cipher image $C_1 = \text{Enc}(P, K_1)$; (c) cipher image $C_2 = \text{Enc}(P, K_2)$; (d) cipher image $C_3 = \text{Enc}(P, K_3)$; (e) difference between $C_2$ and $C_3$, $|C_2 - C_3|$.

FIGURE 11. Key sensitivity analysis in the decryption process. (a) cipher image $C_1$; (b) decrypted result $D_1 = \text{Dec}(C_1, K_1)$; (c) decrypted result $D_2 = \text{Dec}(C_1, K_2)$; (d) decrypted result $D_3 = \text{Dec}(C_1, K_3)$; (e) difference between $D_2$ and $D_3$, $|D_2 - D_3|$.

$$K_2 = \text{ef5bacc3ddb67db619c49e8f2b117}$$
$$\times \text{bcc84250cc0dde10beced4beca405f1c5cc8}$$
$$\times \text{1960f67f5cb89949e47c59eedf}$$
$$\times \text{90a0846ed37ec4d2948981e083e32fc457c83e}$$

$$K_3 = \text{3f5bacc3ddb67db619c49e8f2b}$$
$$\times \text{117bcc84250cc0dde10beced4beca405f1c5cc8}$$
$$\times \text{1960f67f5cb89949e47c59eedf90a}$$
$$\times \text{0846ed37ec4d2948981e083e32fc457c83e}$$

VOLUME 10, 2022
TABLE 8. NPCR and UACI for different encryption schemes (bold indicates test images did not pass the test).

| File name | NPCR | UACI |
|-----------|------|------|
| 256x256   | 0.995693 | 0.3386447 |
| 5.1.09    | 0.99620 | 0.33420 |
| 5.1.10    | 0.99640 | 0.33552 |
| 5.1.11    | 0.99630 | 0.33420 |
| 5.1.12    | 0.99590 | 0.33453 |
| 5.1.13    | 0.99600 | 0.33450 |
| 5.1.14    | 0.99570 | 0.33420 |
| 512x512   | 0.995893 | 0.335541 |
| 5.2.08    | 0.99600 | 0.33420 |
| 5.2.09    | 0.99620 | 0.33454 |
| 5.2.10    | 0.99630 | 0.33510 |
| 7.1.01    | 0.99620 | 0.33465 |
| 7.1.02    | 0.99620 | 0.33465 |
| 7.1.03    | 0.99620 | 0.33465 |
| 7.1.04    | 0.99610 | 0.33465 |
| 7.1.05    | 0.99590 | 0.33465 |
| 7.1.06    | 0.99590 | 0.33465 |
| 7.1.07    | 0.99590 | 0.33465 |
| 7.1.08    | 0.99590 | 0.33465 |
| 7.1.09    | 0.99590 | 0.33465 |
| 7.1.10    | 0.99600 | 0.33465 |
| Boat.512  | 0.99715 | 0.33465 |
| Gray21.512| 0.99643 | 0.33465 |
| Ruler.512 | 0.99637 | 0.33465 |
| 1024x1024 | 0.995994 | 0.33508 |
| 5.3.01    | 0.99620 | 0.33450 |
| 5.3.02    | 0.99620 | 0.33450 |
| 7.2.01    | 0.99620 | 0.33450 |
| Pass rate | 100%  | 100%  |

Figure 10 shows the results of the key sensitivity analysis during encryption. Fig. 10(a) shows the normal image and its histogram, Fig. 10(b), (c), (d) shows the encrypted image with key \( K_1, K_2, K_3 \) and its histogram, respectively, and Fig. 10(e) shows the difference between the two encrypted images. The encrypted images obtained by encrypting the same image with two slightly different keys are entirely different. This indicates that the keys are highly sensitive in the encryption process. Figure 11 shows the results of the key sensitivity analysis during decryption. Fig. 11(a) shows the image encrypted by \( K_1 \) and its histogram, and Fig. 11(b) shows the image recovered by key \( K_1 \). Fig. 11(c) and (d) show the decrypted images by \( K_2 \) and \( K_3 \), respectively. Fig. 11(e) shows the difference between Fig. 11(c) and Fig. 11(d). The image can be recovered only by using the correct key, and even if only one bit is changed to the key, the recovered image will be completely different. Therefore, the key of the encryption scheme proposed in this paper is highly sensitive.

**H. RESISTANCE TO DIFFERENTIAL ATTACKS**

Differential analysis is a common cryptanalysis technique that reveals the effect of small changes in pixels of the plaintext image on the encrypted image. If it means that even small changes in pixels in the plaintext image cause significant differences in the encrypted image, then the algorithm can resist differential attacks well. The pixel change rate (NPCR) and the average change intensity (UACI) are two criteria to test whether the encryption algorithm can resist the differential attack well. We determine whether an encrypted image can pass the NPCR and UACI tests, and strict criteria are given in [65]. The NPCR and UACI are calculated by the following equations, respectively.

\[
D(i, j) = \begin{cases} 
1, & C_1(i, j) \neq C_2(i, j) \\
0, & C_1(i, j) = C_2(i, j) 
\end{cases}
\]

\[
NPCR = \frac{\sum_{i,j} D(i, j)}{W \times H} \times 100\% 
\]

\[
UACI = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |C_1(i, j) - C_2(i, j)| \times 100\%
\]

where \( W, H \) denotes the width and height of the image, \( C_1, C_2 \) denotes two different images, and \( C_1(i, j), C_2(i, j) \) denotes the pixel value of the \((i, j)\) point in each of the two images. We first obtain an encrypted image \( C_1 \) by encrypting
the plaintext image normally, then change the pixel value in a plaintext image to obtain a new encrypted image $C_2$, and then calculate the values of NPCR and UACI. We select images from the USCSIPI “Miscellaneous” dataset for testing, and Table 8 compares the NPCR and UACI values with different algorithms. As shown in Table 8, our proposed algorithm passes the NPCR and UACI tests for all the tested images, so our proposed algorithm can resist various differential attacks well. However, some excellent algorithms fail in several test cases, and our proposed algorithm can show better performance.

Through the above tests, the algorithm proposed in this paper can encrypt the original image well as a meaningless image. Since the proposed algorithm uses a part of the original image as the initial state of the chaotic system, it can resist the selective plaintext attack well. In addition, the proposed algorithm has a strong sensitivity to the key and plaintext image to resist the differential attack, has good security, and can be applied to image encryption.

V. CONCLUSION

This paper proposes an image encryption scheme based on an improved lifting-like structure and cross-plane zigzag transform. To address the problems of the original lifting-like structure and single update function, we modularize the original image and add a new update function and prediction function to obtain a better diffusion effect and security. To address the drawback that the scan mode of the original zigzag transform is limited and can only be used in a single plane, we propose a cross-plane zigzag transform, which increases the security of the overall structure because the mutual replacement between different planes greatly increases the difficulty for crackers to find out the pattern. The simulation results show that our proposed scheme can encrypt different types of digital images into unrecognizable encrypted images. The images can be fully recovered only by using the correct key. The comparison results show that our proposed scheme can better resist various high-quality attacks than other encrypted image algorithms.

In future research work, there are two research ideas. The first is to study the encryption of the structure in the frequency domain because all encryptions of the system are based on the spatial domain, which can significantly increase the complexity of the whole structure when combined with the frequency domain, allowing the security to be improved. The second is to study more complex update and prediction functions to reduce the number of modules of the whole system and increase the encryption efficiency and security. This system can encrypt plaintext images into ciphertext images well, and it is hoped that this system can be widely used in image information security.

REFERENCES

[1] M. Wan, M. Li, G. Yang, S. Gai, and Z. Jin, “Feature extraction using two-dimensional maximum embedding difference,” Inf. Sci., vol. 274, pp. 55–69, Aug. 2014.

[2] 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.

[3] 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., early access, Dec. 27, 2021, doi: 10.1109/TCSVT.2021.3138795.

[4] 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.

[5] 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.

[6] S. M. Seyedzadeh and S. Mirzakuchaki, “A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map,” Signal Process., vol. 92, no. 5, pp. 1202–1215, May 2012.

[7] N. Iqbal, M. Hanif, S. Abbas, M. A. Khan, and Z. U. Rehman, “Dynamic 3D scrambled image based RGB image encryption scheme using hyperchaos and system and DNA encoding,” J. Inf. Secur. Appl., vol. 58, May 2021, Art. no. 102809.

[8] H. Zhou, Q. Zhang, X. P. Wei, and C. J. Zhou, “A summarization on image encryption,” IETE Tech. Rev., vol. 27, no. 6, pp. 503–510, Nov./Dec. 2010.

[9] T. Li and D. Zhang, “Hyperchaotic image encryption based on multiple bit permutation and diffusion,” Entropy, vol. 23, no. 5, Apr. 2021, Art. no. 510.

[10] G. Cheng, C. Wang, and H. Chen, “A novel color image encryption algorithm based on hyperchaos system and permutation-diffusion architecture,” Int. J. Bifurcation Chaos, vol. 29, no. 9, p. 17, Aug. 2019.

[11] R. J. C. Matthews, “On the derivation of a ‘chaotic’ encryption algorithm,” Cryptologia, vol. 13, no. 1, pp. 29–42, Jan. 1989.

[12] J. Fridrich, “Image encryption based on chaotic maps,” in Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Cybern. Simulation, vol. 2, Oct. 1997, pp. 1105–1110.

[13] W. K. Lee, C. W. Phan, W. S. Yap, and B. M. Goh, “Spring: A novel parallel chaos-based image encryption scheme,” Nonlinear Dyn., vol. 92, no. 2, pp. 575–593, Apr. 2018.

[14] X. Huang and G. Ye, “An image encryption algorithm based on irregular wave representation,” Multimed. Tools Appl., vol. 77, pp. 2611–2628, Jan. 2018, doi: 10.1007/s11042-017-4455-x.

[15] Y. Zhou, L. Bao, and C. L. P. Chen, “A new 1D chaotic system for image encryption,” Signal Process., vol. 97, no. 11, pp. 172–182, Apr. 2014.

[16] M. Muñoz-Guillermo, “Image encryption using q-deformed logical map,” Inf. Sci., vol. 552, pp. 352–364, Apr. 2021.

[17] G. Ye, C. Pan, X. Huang, and Q. Mei, “An efficient pixel-level chaotic image encryption algorithm,” Nonlinear Dyn., vol. 94, no. 4, pp. 745–756, Jun. 2018.

[18] 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, May 2021.

[19] G. Hu and B. Li, “A uniform chaotic system with extended parameter range for image encryption,” Nonlinear Dyn., vol. 103, no. 3, pp. 2819–2840, Feb. 2021.

[20] W. Wen, K. Wei, Y. Zhang, Y. Fang, and M. Li, “Colour light field image encryption based on DNA sequences and chaotic systems,” Nonlinear Dyn., vol. 99, no. 2, pp. 1587–1600, Jan. 2020.

[21] Y. Qobbi, A. Jarjar, M. Essaid, and A. Benazzi, “Image encryption algorithm based on genetic operations and chaotic DNA encoding,” Soft Comput., vol. 26, pp. 5823–5832, Jan. 2022.

[22] M. Uldin, F. Jahan, M. K. Islam, and M. Rakib Hassan, “A novel DNA-based key scrambling technique for image encryption,” Complex Intell. Syst., vol. 7, no. 6, pp. 3241–3258, Dec. 2021.

[23] X. Wang and N. Guan, “A novel chaotic image encryption algorithm based on extended Zigzag confusion and RNA operation,” Opt. Laser Technol., vol. 131, Nov. 2020, Art. no. 106366.

[24] A. A. Abbassi, M. Mazinani, and R. Hosseini, “Chaotic evolutionary-based image encryption using RNA codons and amino acid truth table,” Opt. Laser Technol., vol. 132, Dec. 2020, Art. no. 106465.

[25] W. Sirichotedumrong, Y. Kinoshita, and H. Kiya, “Pixel-based image encryption without key management for privacy-preserving deep neural networks,” IEEE Access, vol. 7, pp. 177844–177855, 2019.
