IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 August 2022, accepted 2 September 2022, date of publication 8 September 2022, date of current version 16 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204873

RESEARCH ARTICLE

Fast Chaotic Image Encryption Algorithm Using a Novel Divide and Conquer Diffusion Strategy

BIN GE[1], ZHIHUA SHEN[1], AND JINBAO ZHANG[2]
[1]College of Electronic Information Engineering, Nantong Vocational University, Nantong 226007, China
[2]School of Information Science and Technology, Nantong University, Nantong 226019, China

Corresponding author: Bin Ge (bge@mail.ntvu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62004108; in part by the Natural Science Foundation for Universities of Jiangsu Province under Grant 21KJB520040, Grant 20KJD510001, and Grant 21KJB520041; in part by the Qing Lan Project of Colleges and Universities in Jiangsu Province under Grant 苏教师函 (2020) 10 号; and in part by the Nantong Science and Technology Plan Project under Grant JC2021036 and Grant JC2021147.

ABSTRACT To protect image privacy in real-time transmission, a fast chaotic image encryption algorithm using a novel divide and conquer diffusion strategy is proposed in this paper. Firstly, a fast pseudo-random sequence generator is constructed using 2D hyperchaotic systems. And by performing bitwise XOR operation between two integer sequences obtained from different systems, the final key stream used for diffusion will have better randomness and unpredictability. Secondly, to achieve divide and conquer diffusion, the plain image is divided into three parts, then a parallel CBC-based diffusion method can simultaneously act on upper and bottom parts (or left and right parts), which achieves a time complexity of $\boldsymbol{O}(W + H)$. Moreover, within the diffusion process, the mechanism of information interaction on the central row or column further enhances avalanche effect to resist differential attack. Thirdly, the session key of the proposed algorithm is a mixture of the plain image's hash value and a true random sequence, which not only improve plaintext sensitivity but also realize one-time pad. Finally, experimental results indicate the superiority of our algorithm to resist statistical, chosen-plaintext, entropy, and other common attacks. Furthermore, by comparison with previous works, our algorithm preforms much faster execution speed with only average of 0.08s to encrypt images of size 512×512, while it provide the same or even higher level of security. Therefore, the proposed algorithm can meet security and efficiency requirements of real-time communications of image data.

INDEX TERMS Image encryption, hyperchaotic system, fast encryption, divide and conquer diffusion, one-time pad.

I. INTRODUCTION

With the fast evolution of mobile devices and wireless networks, various types of multimedia data are produced and transmitted on the Internet at a phenomenal rate [1], [2], [3]. As the main form of expression for multimedia, digital images are gaining popularity owing to their advantages of vividness and understandability. However, the openness of the Internet not only provides convenience but also makes it vulnerable to attackers [4], [5]. Since more and more organizations and individuals rely on images to exchange privacy, business, or even military information, the privacy issue of digital images has attracted much concern worldwide [6], [7], [8].

As we all know, encryption techniques are always the most basic tool to protect data security. At present, although block cipher is still the first choice to protect data privacy [9], [10], it confronts fatal flaws when encrypting digital images. The high redundancy of image data will cause heavier time consumption than textual data, let alone invalid encryption may occur now and then due to strong correlations between pixels. Therefore, it is urgent to design specific encryption algorithms for digital images [12], [13], [14], [15]. Since a random permutation of pixel positions can quickly destroy the

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

readability of an image, many scrambling algorithms, such as Arnold transformation [16], Zigzag transformation [17], and magic cube transformation [18], have emerged. But unfortunately, an encrypted image only using scrambling is vulnerable to attackers because it still keeps the statistical properties of the plain image [19], [20]. Hence, a secure image encryption algorithm must convert the image to a noise-like image not only at the visual level but also at the statistical level. Although the application of DNA encoding [21], compressive sensing [22], hash function [23], and other techniques [24] have promoted the progress of image encryption, the chaotic system stands out for constructing a fast and secure cryptosystem due to its strong initial sensitivity and a long period of unpredictability [25], [26], [27], [28].

Since the first report of an algorithm using Logistic map [29] and the maturity of chaos control theory [30], more and more researchers have tried to solve image encryption problems using chaotic systems. In 1998, Fridrich [31] first proposed a general permutation-diffusion structure for chaos-based image encryption algorithms and stressed that the security of algorithm almost depended on the effect of diffusion, which has also been proven by many following cryptanalyses [32], [33], [34]. Although modulo multiplication used in block cipher has proven to be a feasible approach for sufficient diffusion [35], [36], the high time consumption limits its availability in image encryption. Therefore, Chen *et al.* [37], Mao *et al.* [38] employed modulo addition and bitwise XOR (exclusive OR) to propose a more popular diffusion structure based on CBC (cipher block chain). Thus, a pixel is not only encrypted by a secret key but also affected by a previous cipher pixel. Furthermore, to enhance the capability of resistance to differential attacks, many algorithms may apply Chen's diffusion process (or its variants) more than twice by letting the last encrypted pixel of the previous diffusion process be a new initial vector of the following diffusion [39]. Recently, some researchers indicate that the separate execution of permutation and diffusion (whether permutation-then-diffusion or diffusion-then-permutation) may lead to the invalidation of the whole algorithm caused by the weakness of any part. Hence, Liu *et al*. [40] attempted to apply diffusion operation during the row (or column) permutation. On the contrary, Li *et al*. [41] chose to embed pixel scrambling operation into the diffusion process.

However, along with the deepening of research, a growing body of evidence suggests that the CBC structure will also meet efficiency bottlenecks [42], [43]. Typically, to magnify minor changes of any pixel to the whole cipher image, an image with W columns and H rows is normally reshaped to a sequence with length up to $W \times H$, then it will be encrypted pixel by pixel from left to right [37], [38], [39], [41]. As we can see, the execution of traditional CBC-based diffusion process is serial and only handles one pixel at once, so most existing algorithms are unable to cope with real-time secure communications of massive image data. Hence, to overcome this defect, various types of parallel techniques have been introduced to accelerate the diffusion process. Zhao *et al*. [44] segmented the plain image into multiple blocks, then all blocks were concurrently encrypted using a forward-then-backward diffusion strategy. But the lack of diffusion between blocks leaves a potential gap for chosen-plaintext attacks. Wang and Zhao [45] applied the CBC process on matrix rather than on sequence, thus pixels in a row/column could be encrypted in parallel, and by performing a forward-then-reverse diffusion both in vertical and horizontal directions, a cipher image with sufficient diffusion could fast output. However, the problem is that phase-space reconstruction attacks may be a huge threat to this algorithm due to the employed low-dimensional chaotic map. Liu *et al*. [40] not only implemented the CBC process on the matrix but also further enhanced the complexity of diffusion by embedding it with scrambling operation. But only one-round diffusion can hardly guarantee this algorithm to resist chosen-plaintext attacks, while more diffusion round means its higher time consumption.

Therefore, to solve the existing problems of the above-mentioned CBC-based image encryption algorithms, this paper presents a fast chaotic image encryption algorithm using a novel divide and conquer diffusion strategy. First, on basis of a plaintext-related session key calculated by a secure hash function [46], [47], [48], we further mixt it with an external random sequence to obtain a one-time session key. Then, multiple chaotic systems, including two 2D hyperchaotic systems and two cascade chaotic maps, are employed to generate key streams with better cryptographic properties. Afterward, a novel divide and conquer diffusion strategy is proposed to accelerate encryption speed with enough encryption strength. Besides, a corresponding padding method is also presented to meet the image size requirements of the novel diffusion strategy.

The highlights of the present work are summarized below:

(1) A fast chaotic image encryption algorithm using a novel divide and conquer diffusion strategy is proposed. Owing to its high security performance and fast execution speed, the proposed algorithm provides a competitive solution for real-time communications of massive image data.

(2) We construct a fast cryptographic pseudo-random generator consisting of two 2D hyperchaotic systems. Due to its discrete structure but strong chaos properties, the 2D hyperchaotic system iterates much faster than a continuous hyperchaotic system but provides chaotic sequences with the same level of quality [49], [50]. Furthermore, by performing bitwise XOR operations on quantified chaotic sequences, a more uniformly distributed key stream can be fast obtained, which can also enhance its unpredictability and complexity to resist phase-space reconstruction attacks. Finally, the key stream will be reshaped to a $W \times H$ matrix (for an image with the size of $W \times H$) to meet the requirements of the divide and conquer diffusion strategy.

(3) A novel divide and conquer diffusion strategy is designed to accelerate image encryption. The divide

and conquer strategy usually breaks down a big problem into several sub-problems for a simultaneous solution, which can significantly improve computational efficiency. In this paper, we attempt to divide the plain image into two parts for simultaneous diffusion to achieve a time complexity of only $\boldsymbol{O}(W+H)$, while the central row/column acts a more important role. In phase one, the plain image is firstly divided into upper part, bottom part, and central row. Next, in the first round of simultaneous diffusion on two parts, pixels are parallel encrypted row by row. Then, after the initial vector is updated to the central row (it is encrypted by both parts), the second round of simultaneous diffusion can realize sufficient encryption in the vertical direction. At last, after the image is further encrypted by a two-round simultaneous diffusion on the left and right parts, we can obtain a fully encrypted image. Moreover, the embedded permutation operation during diffusion further enhances the encryption strength of the proposed algorithm.

The remainder of this paper is organized as follows. Section II introduces the employed chaotic systems and their roles in this work. The details of the proposed image encryption algorithm are presented in Section III. In Section IV, the applicability and superiority of the proposed image algorithm are demonstrated by experimental results and analyses. Finally, conclusions are drawn in Section V.

II. PRELIMINARY WORKS

The employed chaotic systems are presented in this section, and their roles in our algorithm are also elaborated.

A. TWO 2D HYPERCHAOTIC SYSTEMS

Among various types of chaotic systems, the 2D hyperchaotic system has the advantages of fast iteration speed and strong robustness against degradation. Hence, this paper utilizes two different 2D hyperchaotic systems to produce an unpredictable pseudo-random sequence with a long period and good randomness.

The cross 2D hyperchaotic system (CTDHCS) [49] is defined as

$$\begin{cases} x_1(i+1) = \sin\left(\dfrac{\alpha}{\sin(x_2(i))}\right) \\ x_2(i+1) = \beta \sin\left(\pi\ (x_1(i) + x_2(i))\right) \end{cases} \tag{1}$$

where α and β are two control parameters of the CTDHCS. As shown in Figure 1, when $\alpha = 2$ and $\beta = 1$, two positive Lyapunov exponents appear, indicating that the CTDHCS has evolved into a hyper chaos state. Hence, it can be used as a component to construct a proper pseudo-random generator used for diffusion.

Gao's 2D hyperchaotic system (GTDHCS) [50] is defined as

$$\begin{cases} x_3(i+1) = \sin\left(\dfrac{h\pi}{\sin(x_4(i))}\right) \\ x_4(i+1) = \gamma \sin(\pi x_3(i) x_4(i)) \end{cases} \tag{2}$$

where h and γ are two control parameters of the GTDHCS. It can be seen from Figure 2 that when $h = 5$ and $\gamma = 5$, the GTDHCS demonstrates hyper chaos properties since it now contains two positive Lyapunov exponents. Therefore, the GTDHCS can be utilized as another component of secure pseudo-random generator to produce diffusion keys.

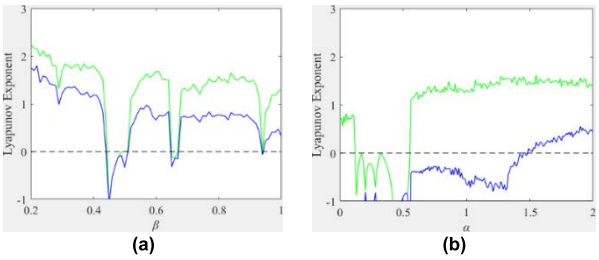


FIGURE 1. Lyapunov exponents spectrum of GTDHCS, (a) $\alpha = 2$ and $\beta \in (0, 1]$, (b) $\alpha = (0, 2]$ and $\beta = 1$.

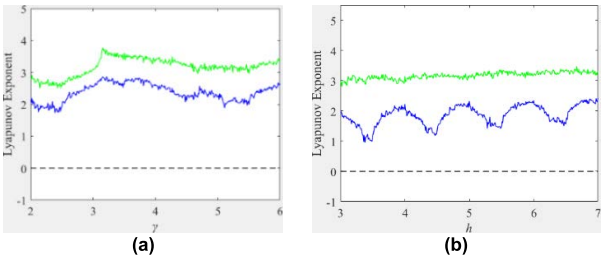


FIGURE 2. Lyapunov exponents spectrum of CTDHCS, (a) $h = 5$ and $\gamma \in (0, 5]$, (b) $h \in (0, 5]$ and $\gamma = 5$.

B. TWO CASCADE CHAOTIC MAPS

In 2021, by using a universal method, Yuan *et al.* [51] constructed a series of cascade chaotic maps to provide better randomness and stronger robustness for image encryption.

To fast and sufficiently scramble pixel's positions, our algorithm employs the cascade Logistic-Logistic chaotic map (CLLCM), which is given by

$$y_1(i+1) = 1 - 2(1 - 2y_1(i)^2)^2 \tag{3}$$

Meanwhile, our algorithm utilizes the cascade Logistic-Sine chaotic map (CLSCM), as shown in Equation (4), to produce an initial vector with high-quality for the CBC-based diffusion process.

$$y_2 = 4\sin(\pi x_i)(1 - \sin(\pi x_i)) \tag{4}$$

As demonstrated in Figure 3, the largest Lyapunov exponents of CLLCM and CLSCM are greater than the original Logistic map or Sine map, indicating that they can provide a better random number for our algorithm.

III. THE PROPOSED IMAGE ENCRYPTION ALGORITHM

The proposed algorithm is detailed in this section, including 1) The generation method of a plaintext-related one-time session key; 2) The padding strategy of plain image;

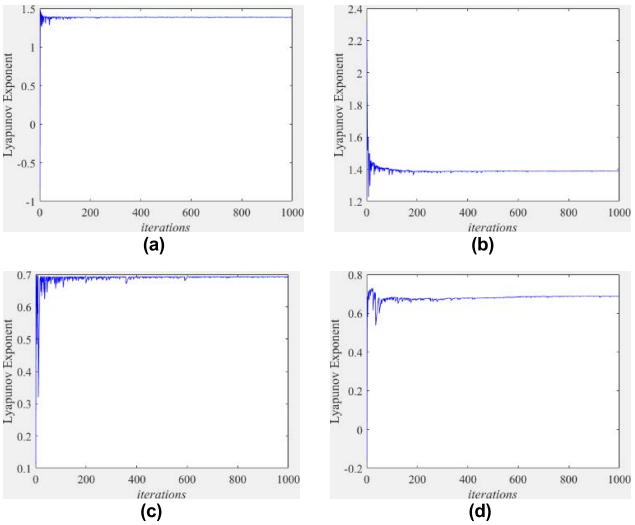


FIGURE 3. Largest Lyapunov exponents of (a) CLLCM, (b) CLSCM, (c) Logistic map, and (d) Sine map.

3) The generation method of initial vector; 4) The generation method of diffusion keys; 5) The generation method of permutation keys; 6) The divide and conquer diffusion based encryption process; 7) The decryption process. Figure 4 exhibits the flow diagram of encryption end.

A. THE GENERATION METHOD OF A PLAINTEXT-RELATED ONE-TIME SESSION KEY

Establishing associations between the plain image and session key is the most effective approach for enhancing the plaintext sensitivity of a chaos-based encryption algorithm.

Moreover, this paper firstly performs SHA-256 on a plain image to obtain its hash value, then further mixes the result with a true random sequence to obtain a plaintext-related one-time session key, and ultimately extracts the initial values of chaotic maps. The steps are detailed below:

Step A-1: Input the plain image into the SHA-256 function to generate its hash value $\boldsymbol{H}$, which is a hexadecimal sequence with a length of 64.

Step A-2: Mix $\boldsymbol{H}$ with an external key $\boldsymbol{EK}$ (it consists of 64 true random hexadecimal numbers) using $\boldsymbol{SK}$ = bitxor ($\boldsymbol{H}$, $\boldsymbol{EK}$) to obtain the session key, where bitxor represents a bitwise exclusive-OR operation.

Step A-3: Extract the initial values for the above-mentioned chaotic systems by

$$\begin{cases} x_1(1) = \text{hex2dec}(\boldsymbol{SK}(1:12))/2^{48} \\ x_2(1) = \text{hex2dec}(\boldsymbol{SK}(10:21))/2^{48} \\ x_3(1) = \text{hex2dec}(\boldsymbol{SK}(19:30))/2^{48} \\ x_4(1) = \text{hex2dec}(\boldsymbol{SK}(28:39))/2^{48} \\ y_1(1) = \text{hex2dec}(\boldsymbol{SK}(37:48))/2^{48} \\ y_2(1) = \text{hex2dec}(\boldsymbol{SK}(46:57))/2^{48} \end{cases} \tag{5}$$

where the hex2dec operation can convert the hexadecimal number to a decimal number, then $x_1(1)$ and $x_2(1)$ are initial states of CTDHCS, $x_3(1)$ and $x_4(1)$ are initial states of GTDHCS, $y_1(1)$ and $y_2(1)$ are initial states of CLLCM and CLSCM respectively.

Step A-4: To overcome the security risks caused by transient effect, pre-iteration is necessary for the chaotic system used in encryption algorithms. Since pre-iteration also affects the encryption result, this paper handles the remaining part of $\boldsymbol{H}$ by

$$\begin{cases} p_1 = \text{bitxor}(\text{hex2dec}(\boldsymbol{H}(58:60)), \\ \quad \text{hex2dec}(\boldsymbol{H}(49:51))) + 20 \\ p_2 = \text{bitxor}(\text{hex2dec}(\boldsymbol{H}(60:62)), \\ \quad \text{hex2dec}(\boldsymbol{H}(52:54))) + 20 \\ p_3 = \text{bitxor}(\text{hex2dec}(\boldsymbol{H}(62:64)), \\ \quad \text{hex2dec}(\boldsymbol{H}(55:57))) + 20 \end{cases} \tag{6}$$

to obtain the pre-iteration p_1 for CTDHCS and GTDHCS, p_2 for CLLCM, and p_3 for CLSCM.

B. THE PADDING STRATEGY OF PLAIN IMAGE

To achieve the divide and conquer diffusion strategy within our encryption process, the plain image must be divided into three parts: left half, central column, and right half when encryption in the horizontal direction; or upper half, bottom half, and central row when encryption in the vertical direction. It indicates that the plain image should have an odd number of rows and columns. Since not all images meet this requirement, this paper applies **Algorithm 1** to pad a $W \times H$ image if necessary, and the details are as follows:

C. THE GENERATION METHOD OF DIFFUSION KEYS

To obtain enough random keys for the proposed diffusion strategy, this paper inputs the extracted parameters into CTDHCS and GTDHCS and then employs **Algorithm 2**to construct a key matrix $\boldsymbol{K}_{\text{d}}$ with the same size as $\boldsymbol{P}$.

D. THE GENERATION METHOD OF PERMUTATION KEYS

To realize high-strength encryption, simultaneous scrambling operations are performed during the diffusion process. Thus, we employ CLLCM to randomly control the circular shift of row and column, then the keys for scrambling are produced by the following steps:

Step D-1: Input $y_1(1)$, p_2, H, and W into CLLCM.

Step D-2: Iterate CLLCM p_2 times, but drop the state value.

Step D-3: Initialize an empty vector $\boldsymbol{V}$.

Step D-4: Continue to iterate CLLCM, and let $\boldsymbol{V} = \{\boldsymbol{V}, y_1(i)\}, i = p_2+1, p_2+2, \ldots, p_2+2H+2W-1, p_2+2H+2W$.

Step D-5: Obtain permutation keys $\boldsymbol{K}_{\text{pr}}$ and $\boldsymbol{K}_{\text{pc}}$ for rows and columns by Equation (7), and then drop $\boldsymbol{V}$.

$$\begin{cases} \boldsymbol{K}_{\text{pr1}} = \left\lfloor \boldsymbol{V}(1:H) \times 10^{15} \right\rfloor \%W \\ \boldsymbol{K}_{\text{pc1}} = \left\lfloor \boldsymbol{V}(H+1:H+W) \times 10^{15} \right\rfloor \%H \\ \boldsymbol{K}_{\text{pr2}} = \left\lfloor \boldsymbol{V}(H+W+1:2H+W) \times 10^{15} \right\rfloor \%W \\ \boldsymbol{K}_{\text{pr2}} = \left\lfloor \boldsymbol{V}(2H+W+1:2H+2W) \times 10^{15} \right\rfloor \%H \end{cases} \tag{7}$$

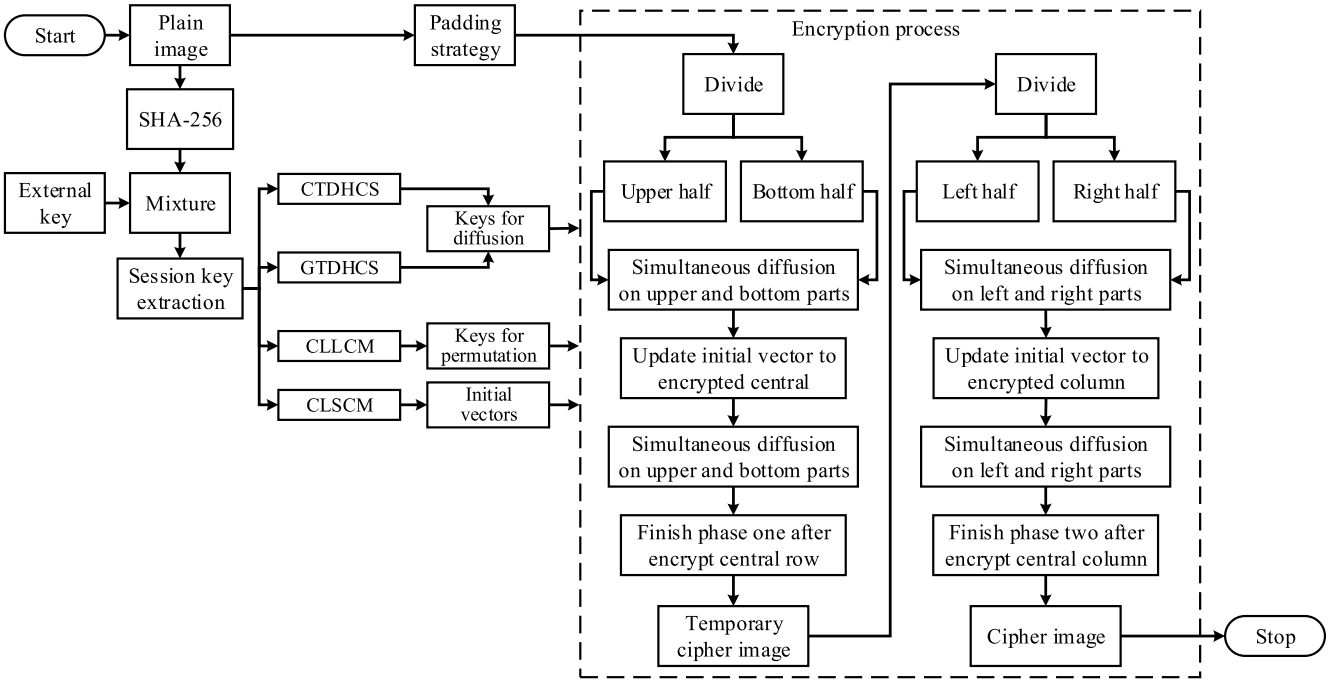


FIGURE 4. Flow diagram of encryption.

Algorithm 1 The Padding Strategy of Plain Image

Input: The original plain image $\boldsymbol{P}$

Output: The updated $\boldsymbol{P}$, H, W after padding

$[H, W] \leftarrow$ size($\boldsymbol{P}$)
$\mathbf{L} \leftarrow$ {101,118,101,110,69,86,69,78}
/∗ **L** is a label consisting of the decimal numbers of ASCII codes with e: 101, v: 118, n: 110, E: 69, V: 86, and N: 78. Therefore, **L**represents {E, V, E, N, e, v, e, n}, which provides a basis for correctly removing padding after decryption. ∗/
if $W\%2 == 0$
  Initialize a vector $\boldsymbol{V}$ with a height of H with all elements being zero
  $\boldsymbol{V}(1{:}8) \leftarrow \mathbf{L}^{\mathrm{T}}$// T represents the transpose operation
  $\boldsymbol{V}(H\text{-}7{:}H) \leftarrow \mathbf{L}^{\mathrm{T}}$
  $\boldsymbol{P} \leftarrow \{\boldsymbol{P}(:,1{:}W/2), \boldsymbol{V}, \boldsymbol{P}(:,W/2{+}1{:}W)\}$
  $W \leftarrow W + 1$
  Drop $\boldsymbol{V}$
end if
if $H\%2 ==0$
  Initialize a vector $\boldsymbol{V}$ with a width of W with all elements being zero
  $\boldsymbol{V}(1{:}8) \leftarrow \mathbf{L}$
  $\boldsymbol{V}(W\text{-}7{:}W) \leftarrow \mathbf{L}$
  $\boldsymbol{P} \leftarrow \{\boldsymbol{P}(1{:}H/2,:); \boldsymbol{V}; \boldsymbol{P}(H/2{+}1{:}H,:)\}$
  $H \leftarrow H + 1$
  Drop $\boldsymbol{V}$
end if

E. THE GENERATION METHOD OF INITIAL VECTOR

An initial vector will also influence the effect of CBC-based encryption. Therefore, we adopt on the following steps to fast create a random initial vector according to the size of the plain image in this paper:

Step E-1: Input $y_2(1)$, $p3$, and H into CLSCM.

Step E-2: Iterate CLSCM p_3 times, but drop the state value.

Step E-3: Initialize an empty vector $\boldsymbol{V}$.

Step E-4: Continue to iterate CLSCM, and let $\boldsymbol{V} = \{\boldsymbol{V}, y_2(i)\}$, $i = p_3 + 1, p_3 + 2, \ldots, p_3 + W\text{-}1, p_3 + W$.

Step E-5: Obtain initial vector $\boldsymbol{I}$ by $\boldsymbol{I} = \lfloor \boldsymbol{V} \times 10^{15} \rfloor \%256$, then drop $\boldsymbol{V}$.

F. THE DIVIDE AND CONQUER DIFFUSION BASED ENCRYPTION PROCESS

Now, we can perform **Algorithm 3** to encrypt a plain image, and Figure 5 illustrates how the pixel information is fast and sufficiently diffused to the whole cipher image by the proposed divide and conquer diffusion strategy.

Figure 5 demonstrates a demo of the encryption process consisting of two main stages: 1) Parallel encryption in the vertical direction. 2) Parallel encryption in the horizontal direction.

As illustrated in Figure 5 (a), in the first stage, the plain image is divided into the upper part, bottom part, and central row, then the pixels are parallel encrypted row by row from both upper and bottom parts. Meanwhile, diffusion is realized by applying the previous encrypted row to calculate the ciphertext of the next row. Moreover, the complexity and security of the proposed encryption algorithm are further

Algorithm 2 The Generation Method of Diffusion Keys

Input: $x_1(1), x_2(1), x_3(1), x_4(1), p_1, H, W$
Output: $\boldsymbol{K}_{\mathrm{d}}$
//pre-iterations to eliminate the transient effect
for i from 1 to p_1
  $x_1(i+1) \leftarrow \sin(\alpha/\sin(x_2(i)))$
  $x_2(i+1) \leftarrow \beta \times \sin(\pi \times (x_1(i)+x_2(i)))$
  $x_3(i+1) \leftarrow \sin(h \times \pi/\sin(x_4(i)))$
  $x_4(i+1) \leftarrow \gamma \times \sin(\pi \times x_3(i) \times x_4(i))$
end for
//update $x_1(1), x_2(1), x_3(1), x_4(1)$
$x_1(1) \leftarrow x_1(p_1{+}1), x_2(1) \leftarrow x_2(p_1{+}1)$, $x_3(1) \leftarrow x_3(p_1{+}1), x_4(1) \leftarrow x_4(p_1{+}1)$
Initialize two empty vectors $\boldsymbol{V}_1, \boldsymbol{V}_2$
for i from 1 to $\lceil W \times H/2 \rceil$-1
  $x_1(i+1) \leftarrow \sin(\alpha/\sin(x_2(i)))$
    $x_2(i+1) \leftarrow \beta \times \sin(\pi \times (x_1(i)+x_2(i)))$
    $x_3(i+1) \leftarrow \sin(h \times \pi/\sin(x_4(i)))$
  $x_4(i+1) \leftarrow \gamma \times \sin(\pi \times x_3(i) \times x_4(i))$
  $\boldsymbol{V}_1 \leftarrow \{\boldsymbol{V}_1, x_1(i), x_2(i)\}$
  $\boldsymbol{V}_2 \leftarrow \{\boldsymbol{V}_2, x_4(i), x_3(i)\}$
end for
Initialize empty vectors $\boldsymbol{T}_1, \boldsymbol{T}_2$
$\boldsymbol{T}_1 \leftarrow \lfloor(|\boldsymbol{V}_1(1{:}W \times H)| - \lfloor|\boldsymbol{V}_1(1{:}W \times H)|\rfloor) \times 10^{15}\rfloor \% 256$
$\boldsymbol{T}_2 \leftarrow \lfloor(|\boldsymbol{V}_1(1{:}W \times H)| - \lfloor|\boldsymbol{V}_1(1{:}W \times H)|\rfloor) \times 10^{15}\rfloor \% 256$
Drop $\boldsymbol{V}_1, \boldsymbol{V}_2$ and then initialize an empty vector $\boldsymbol{V}$
$\boldsymbol{V} \leftarrow \mathrm{bitxor}(\boldsymbol{T}_1, \boldsymbol{T}_2)$
$\boldsymbol{K}_{\mathrm{d}} \leftarrow \mathrm{reshape}(\boldsymbol{V}, H, W)$ then drop $\boldsymbol{V}$

enhanced by interleaving the random row circular shift in the diffusion process. The central row plays an important role in information interaction between the upper part and bottom part. By encrypting the central row using the ciphertext of both the upper and bottom parts and taking it as a new initial vector, diffusion between the upper part and bottom part can be implemented to finish sufficient encryption in the vertical direction.

Next, as shown in Figure 5(b), to meet the requirement of avalanche effect, parallel encryption in the horizontal direction is essential to make the whole cipher image influenced by minor changes of a pixel. Thus, the temporary cipher image created by the first stage is divided into the left part, right part, and central column. Firstly, the central column is used as the initial vector to connect the first stage and the second stage. Then, the pixels are parallel encrypted column by column from both the left and right parts. Similar to the first stage, the central column plays an important role in achieving sufficient encryption in the vertical direction.

Algorithm 3 The Encryption Process

Input: $\boldsymbol{K}_{\mathrm{d}}, \boldsymbol{K}_{\mathrm{pr1}}, \boldsymbol{K}_{\mathrm{pc1}}, \boldsymbol{K}_{\mathrm{pr2}}, \boldsymbol{K}_{\mathrm{pc2}}, \boldsymbol{I}, \boldsymbol{P}$
Output: $\boldsymbol{C}$
Initialize a $W \times H$ Matrix $\boldsymbol{TC}$ with all elements being zero
/*Start diffusion in the vertical direction using $\boldsymbol{I}$, $\boldsymbol{K}_{\mathrm{d}}$, and $\boldsymbol{K}_{\mathrm{pr1}}$ simultaneously from the upper part and bottom part*/
$\boldsymbol{Temp}_1 \leftarrow (\boldsymbol{I} + \boldsymbol{P}(1,:))\%256$
$\boldsymbol{Temp}_2 \leftarrow (\boldsymbol{I} + \boldsymbol{P}(H,:))\%256$
$\boldsymbol{Temp}_1 \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}(1,:), \boldsymbol{Temp}_1)$
$\boldsymbol{Temp}_2 \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}(H,:), \boldsymbol{Temp}_2)$
$\boldsymbol{TC}(1,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}_1, \boldsymbol{K}_{\mathrm{pr1}}(1))$
$\boldsymbol{TC}(H,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}_2, \boldsymbol{K}_{\mathrm{pr1}}(H))$
/*Continue to diffuse the remaining rows simultaneously from the upper part and bottom part*/
for i from 2 to $(H\text{-}1)/2$
  $\boldsymbol{Temp}_1 \leftarrow (\boldsymbol{TC}(i\text{-}1,:) + \boldsymbol{P}(i,:))\%256$
  $\boldsymbol{Temp}_2 \leftarrow (\boldsymbol{TC}(H+2\text{-}i,:) + \boldsymbol{P}(H+1\text{-}i,:))\%256$
  $\boldsymbol{Temp}_1 \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}(1,:), \boldsymbol{Temp}_1)$
  $\boldsymbol{Temp}_2 \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}(H{+}1\text{-}i,:), \boldsymbol{Temp}_2)$
  $\boldsymbol{TC}(i,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}_1, \boldsymbol{K}_{\mathrm{pr1}}(i))$
  $\boldsymbol{TC}(H+1\text{-}i,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}_2, \boldsymbol{K}_{\mathrm{pr1}}(H{+}1\text{-}i))$
end for
//Information interaction on the central row
$\boldsymbol{Temp} \leftarrow (\boldsymbol{TC}((H\text{-}1)/2,:) + \boldsymbol{TC}((H\text{-}1)/2{+}2,:) + \boldsymbol{P}((H\text{-}1)/2{+}1,:))\%256$
$\boldsymbol{Temp} \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}((H\text{-}1)/2{+}1,:), \boldsymbol{Temp})$
$\boldsymbol{TC}((H\text{-}1)/2{+}1,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}, \boldsymbol{K}_{\mathrm{pr1}}((H\text{-}1)/2{+}1))$
/*Repeat the above diffusion steps using $\boldsymbol{K}_{\mathrm{d}}$, $\boldsymbol{K}_{\mathrm{pr2}}$, and the updated $\boldsymbol{I}$
$\boldsymbol{I} \leftarrow \boldsymbol{TC}((H\text{-}1)/2{+}1,:)$
$\boldsymbol{Temp}_1 \leftarrow (\boldsymbol{I} + \boldsymbol{TC}(1,:))\%256$
21: $\boldsymbol{Temp}_2 \leftarrow (\boldsymbol{I} + \boldsymbol{TC}(H,:))\%256$
$\boldsymbol{Temp}_1 \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}(1,:), \boldsymbol{Temp}_1)$
$\boldsymbol{Temp}_2 \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}(H,:), \boldsymbol{Temp}_2)$
$\boldsymbol{TC}(1,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}_1, \boldsymbol{K}_{\mathrm{pr2}}(1))$
$\boldsymbol{TC}(H,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}_2, \boldsymbol{K}_{\mathrm{pr2}}(H))$
for i from 2 to $(H\text{-}1)/2$
  $\boldsymbol{Temp}_1 \leftarrow (\boldsymbol{TC}(i\text{-}1,:) + \boldsymbol{TC}(i,:))\%256$
  $\boldsymbol{Temp}_2 \leftarrow (\boldsymbol{TC}(H+2\text{-}i,:) + \boldsymbol{TC}(H+1\text{-}i,:))\%256$
  $\boldsymbol{Temp}_1 \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}(1,:), \boldsymbol{Temp}_1)$
  $\boldsymbol{Temp}_2 \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}(H{+}1\text{-}i,:), \boldsymbol{Temp}_2)$
  $\boldsymbol{TC}(i,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}_1, \boldsymbol{K}_{\mathrm{pr2}}(i))$
  $\boldsymbol{TC}(H+1\text{-}i,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}_2, \boldsymbol{K}_{\mathrm{pr2}}(H{+}1\text{-}i))$
end for
//Information interaction on central row
$\boldsymbol{Temp} \leftarrow (\boldsymbol{TC}((H\text{-}1)/2,:) + \boldsymbol{TC}((H\text{-}1)/2{+}2,:) + \boldsymbol{TC}((H\text{-}1)/2{+}1,:))\%256$
$\boldsymbol{Temp} \leftarrow \mathrm{bitxor}(\boldsymbol{K}_{\mathrm{d}}((H\text{-}1)/2{+}1,:), \boldsymbol{Temp})$
$\boldsymbol{TC}((H\text{-}1)/2{+}1,:) \leftarrow \mathrm{circshift}(\boldsymbol{Temp}, \boldsymbol{K}_{\mathrm{pr2}}((H\text{-}1)/2{+}1))$
/*Start diffusion in the horizontal direction using $\boldsymbol{K}_{\mathrm{d}}$, $\boldsymbol{K}_{\mathrm{pc1}}$, and the updated $\boldsymbol{I}$ simultaneously from the left part and right part*/

Algorithm 3 *(Continued.)* The Encryption Process

$\boldsymbol{I} \leftarrow \boldsymbol{TC}(:, (W\text{-}1)/2+1)$
$\boldsymbol{Temp}_1 \leftarrow (\boldsymbol{I} + \boldsymbol{TC}(:, 1))\%256$
$\boldsymbol{Temp}_2 \leftarrow (\boldsymbol{I} + \boldsymbol{TC}(:, H))\%256$
$\boldsymbol{Temp}_1 \leftarrow \text{bitxor}(\boldsymbol{K}_{\text{d}}(:, 1), \boldsymbol{Temp}_1)$
$\boldsymbol{Temp}_2 \leftarrow \text{bitxor}(\boldsymbol{K}_{\text{d}}(:, H), \boldsymbol{Temp}_2)$
$\boldsymbol{TC}(:, 1) \leftarrow \text{circshift}(\boldsymbol{Temp}_1, \boldsymbol{K}_{\text{pr1}}(1))$
$\boldsymbol{TC}(:, H) \leftarrow \text{circshift}(\boldsymbol{Temp}_2, \boldsymbol{K}_{\text{pr1}}(H))$
/∗Continue to diffuse the remaining rows simultaneously from the left part and right part∗/
for i from 2 to $(W\text{-}1)/2$
  $\boldsymbol{Temp}_1 \leftarrow (\boldsymbol{TC}(:, i\text{-}1) + \boldsymbol{TC}(:, i))\%256$
  $\boldsymbol{Temp}_2 \leftarrow (\boldsymbol{TC}(:, W + 2\text{-}i) + \boldsymbol{TC}(:, W + 1\text{-}i))\%256$
  $\boldsymbol{Temp}_1 \leftarrow \text{bitxor}(\boldsymbol{K}_{\text{d}}(:, 1), \boldsymbol{Temp}_1)$
  $\boldsymbol{Temp}_2 \leftarrow \text{bitxor}(\boldsymbol{K}_{\text{d}}(:, W + 1\text{-}i), \boldsymbol{Temp}_2)$
  $\boldsymbol{TC}(:, i) \leftarrow \text{circshift}(\boldsymbol{Temp}_1, \boldsymbol{K}_{\text{pc1}}(i))$
  $\boldsymbol{TC}(:, W + 1\text{-}i) \leftarrow \text{circshift}(\boldsymbol{Temp}_2, \boldsymbol{K}_{\text{pc1}}(W+1\text{-}i))$
end for
//Information interaction on central column
$\boldsymbol{Temp} \leftarrow (\boldsymbol{TC}(:, (W\text{-}1)/2)+\boldsymbol{TC}(:, (W\text{-}1)/2+2)+\boldsymbol{TC}(:, (H\text{-}1)/2+1))\%256$
$\boldsymbol{Temp} \leftarrow \text{bitxor}(:, \boldsymbol{K}_{\text{d}}((W\text{-}1)/2+1), \boldsymbol{Temp})$
$\boldsymbol{TC}(:, (W\text{-}1)/2+1) \leftarrow \text{circshift}(\boldsymbol{Temp}, \boldsymbol{K}_{\text{pc1}}((W\text{-}1)/2+1))$
/∗Repeat the above diffusion steps using $\boldsymbol{K}_{\text{d}}$, $\boldsymbol{K}_{\text{pc2}}$, and the updated $\boldsymbol{I}$
$\boldsymbol{I} \leftarrow \boldsymbol{TC}(:, (W\text{-}1)/2+1)$
$\boldsymbol{Temp}_1 \leftarrow (\boldsymbol{I} + \boldsymbol{TC}(:, 1))\%256$
$\boldsymbol{Temp}_2 \leftarrow (\boldsymbol{I} + \boldsymbol{TC}(:, H))\%256$
$\boldsymbol{Temp}_1 \leftarrow \text{bitxor}(\boldsymbol{K}_{\text{d}}(:, 1), \boldsymbol{Temp}_1)$
$\boldsymbol{Temp}_2 \leftarrow \text{bitxor}(\boldsymbol{K}_{\text{d}}(:, H), \boldsymbol{Temp}_2)$
$\boldsymbol{TC}(:, 1) \leftarrow \text{circshift}(\boldsymbol{Temp}_1, \boldsymbol{K}_{\text{pr1}}(1))$
$\boldsymbol{TC}(:, H) \leftarrow \text{circshift}(\boldsymbol{Temp}_2, \boldsymbol{K}_{\text{pr1}}(H))$
for i from 2 to $(W\text{-}1)/2$
  $\boldsymbol{Temp}_1 \leftarrow (\boldsymbol{TC}(:, i\text{-}1) + \boldsymbol{TC}(:, i))\%256$
  $\boldsymbol{Temp}_2 \leftarrow (\boldsymbol{TC}(:, W + 2\text{-}i) + \boldsymbol{TC}(:, W + 1\text{-}i))\%256$
  $\boldsymbol{Temp}_1 \leftarrow \text{bitxor}(\boldsymbol{K}_{\text{d}}(:, 1), \boldsymbol{Temp}_1)$
  $\boldsymbol{Temp}_2 \leftarrow \text{bitxor}(\boldsymbol{K}_{\text{d}}(:, W + 1\text{-}i), \boldsymbol{Temp}_2)$
  $\boldsymbol{TC}(:, i) \leftarrow \text{circshift}(\boldsymbol{Temp}_1, \boldsymbol{K}_{\text{pc1}}(i))$
  $\boldsymbol{TC}(:, W + 1\text{-}i) \leftarrow \text{circshift}(\boldsymbol{Temp}_2, \boldsymbol{K}_{\text{pc1}}(W+1\text{-}i))$
end for
//Information interaction on central column
$\boldsymbol{Temp} \leftarrow (\boldsymbol{TC}(:, (W\text{-}1)/2)+\boldsymbol{TC}(:, (W\text{-}1)/2+2)+\boldsymbol{TC}(:, (H\text{-}1)/2+1))\%256$
$\boldsymbol{Temp} \leftarrow \text{bitxor}(:, \boldsymbol{K}_{\text{d}}((W\text{-}1)/2+1), \boldsymbol{Temp})$
$\boldsymbol{TC}(:, (W\text{-}1)/2+1) \leftarrow \text{circshift}(\boldsymbol{Temp}, \boldsymbol{K}_{\text{pc2}}((W\text{-}1)/2+1))$
$\boldsymbol{C} \leftarrow \boldsymbol{TC}$//Finally the cipher image $\boldsymbol{C}$ is obtained

In general, the proposed divide and conquer diffusion strategy achieves high security by full encryption in both vertical and horizontal directions, while it provides a high operating efficiency with a time complexity of only $\boldsymbol{O}(W + H)$.

Algorithm 4 The Removing Padding Strategy of Decrypted Image

Input: The decrypted image $\boldsymbol{D}$
Output: The original plain image $\boldsymbol{P}$

$[H, W] \leftarrow \text{size}(\boldsymbol{D})$
$\mathbf{L} \leftarrow \{101,118,101,110,69,86,69,78\}$
// Remove the inserted row if needed
if$\text{bitand}(\boldsymbol{D}((H\text{-}1)/2+1, 1:8), \boldsymbol{D}((H\text{-}1)/2+1, W\text{-}7:W)) == \mathbf{L}$
  $\boldsymbol{D}' \leftarrow \{\boldsymbol{D}(1:(H\text{-}1)/2, :); \boldsymbol{D}((H\text{-}1)/2+2:H, :)\}$
  $H \leftarrow H - 1$
end if
// Remove the inserted column if needed
if$\text{bitand}(1:8, \boldsymbol{D}'((W\text{-}1)/2+1), \boldsymbol{D}'(H\text{-}7:H, (W\text{-}1)/2+1)) == \mathbf{L}^{\text{T}}$
  $\boldsymbol{P} \leftarrow \{\boldsymbol{D}'(:, 1:(W\text{-}1)/2), \boldsymbol{D}'(:, (W\text{-}1)/2+2):W\}$
  $W \leftarrow W - 1$
end if

G. THE DECRYPTION PROCESS

Since gray values are in the range of 0-255, all operations in **Algorithm 3** have valid inverse operations. Hence, as shown in Figure 6, the original plain image can be recovered at decryption end using a correct session key.

However, due to the padding strategy before the encryption process, the decrypted image is not always the original plain image. Therefore, **Algorithm 4** must be applied to remove the central row or column based on the labels inserted by padding.

IV. EXPERIMENTAL RESULTS AND ANALYSES

The applicability of the proposed algorithm is evaluated by different kinds of tests in this section. Indeed, the analyses and comparisons can demonstrate the superior security and efficiency of our algorithm. All tests are performed on the MATLAB R2016b platform using a personal computer equipped with Intel(R) Core (TM) i5 6500 CPU @ 3.20GHz, 16 GB 2400Hz DDR4, 256 GB M.2 SSD, and running a 64-bit Window 7 operating system (professional edition).

For simulation, we access Random.org [52] to obtain a true random external key $\boldsymbol{EK}$ = {4c9df03c6c4ca7ead0b00b62069bde031799dcc44ee24bc572ff49b931c26253}.

Several standard 8-bit gray images of the USC-SIPI image database are used as test objects: 1) 'Lena' of size 256×256; 2) 'Resolution' of size 256×256; 3) 'Baboon' of size 512×512; 4) 'Ruler' of size 512×512; 5) 'Airplane' of size 1024×1024; 6) 'Male' of size 1024×1024. Their encryption and decryption results are presented in Figure 7.

The comparison of Figure 7 (column one) and Figure 7 (column two) indicates that all plain images containing rich information are successfully encrypted into totally disordered images. Then, as shown in Figure 7 (column three), all cipher images can be recovered to original images by the decryption

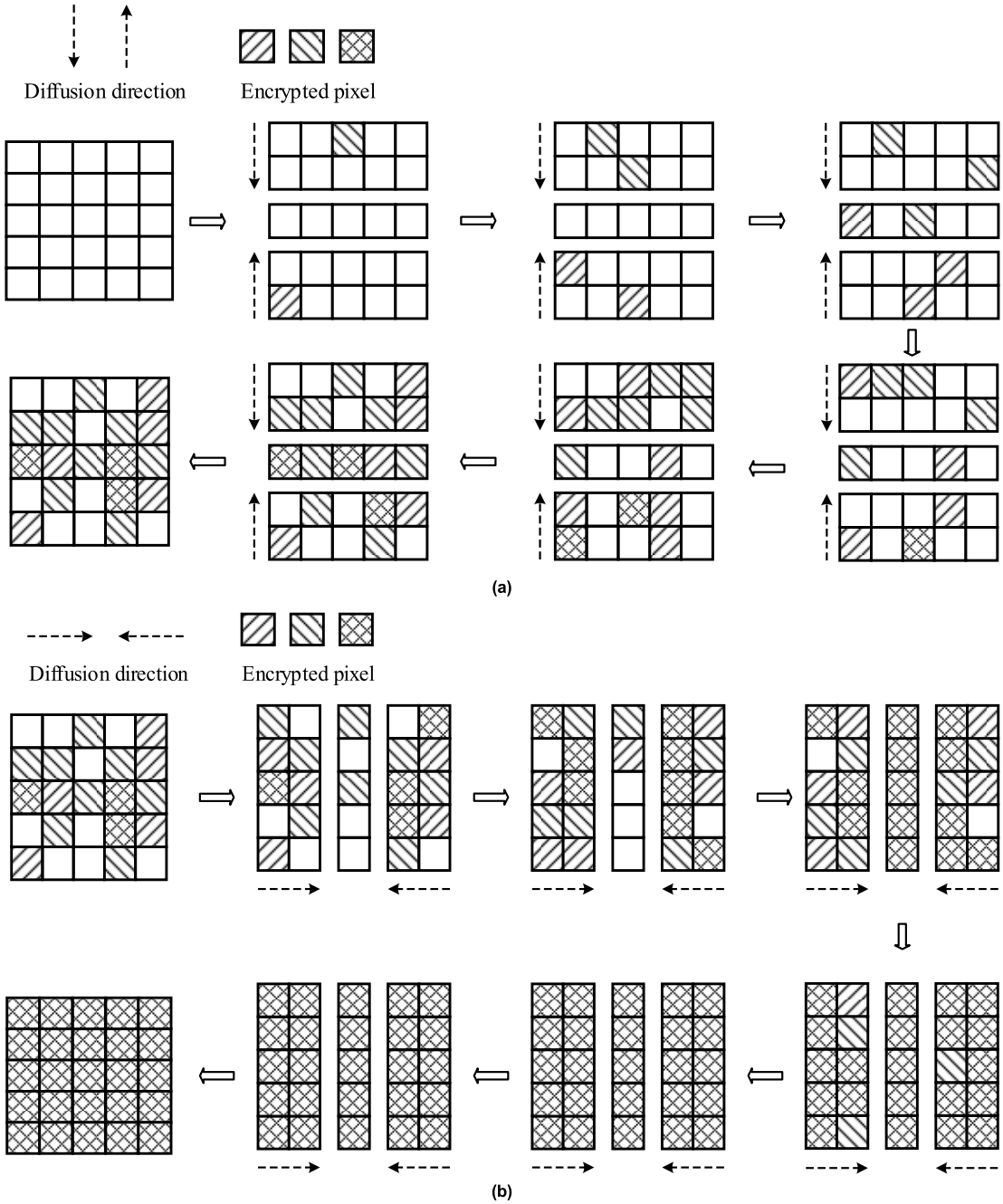


FIGURE 5. Demo of the encryption process. The effect of diffusion (a) in the vertical direction, and (b) in the horizontal direction.

process using their corresponding session keys, which confirm the validity of the proposed algorithm.

A. HISTOGRAM ANALYSIS

The histogram is an important visual tool for observing the distribution of an image's pixels. A valid image encryption algorithm should output a uniformly distributed cipher image to resist frequency attacks. Figure 8 demonstrates the results of histogram tests. The fluctuant histograms of plain images in Figure 8 (column two) indicate that each image contains different and rich information. Then, in the fourth column, all the histograms are approximately flat, showing that our algorithm can successfully hide the distributions of the original images.

Furthermore, based on the histogram, the chi-square (χ^2) test [40] defined in Equation (8) is a vital tool for deep quantitative analysis of the flatness of the histogram.

$$\chi^2 = \sum_{i=1}^{L} \frac{(f_i - p_i)^2}{p_i} \tag{8}$$

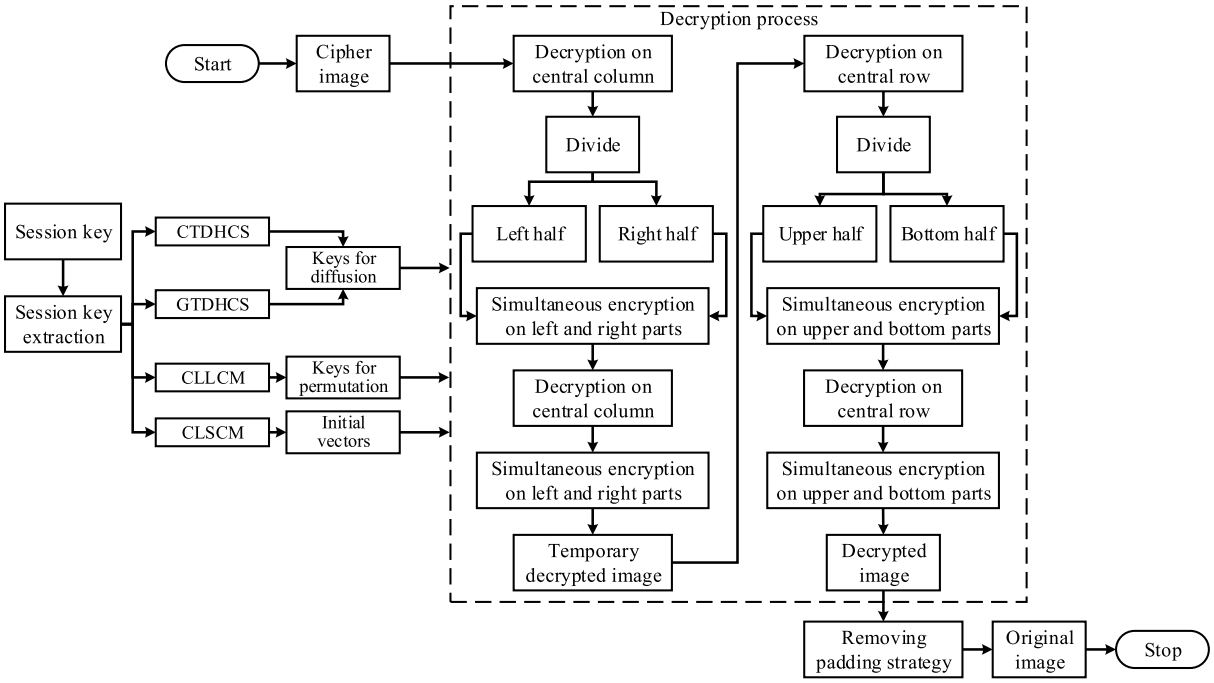


FIGURE 6. Flow diagram of the decryption.

where L represents the grey level (for an 8-bit grayscale image, its maximum level $L = 256$), p_i and f_i are respectively the expected proportion and real proportion of each pixel value in a test image. For a uniform distributed cipher image, it should have an expected $\chi^2 = 293.24783$ at the significance level $\alpha = 0.05$. Since all the χ^2 values of the above cipher images in Table 1 are less than 293.24783, the corresponding P-values are far more than 0.05. The results indicate that our algorithm can achieve the goal to protect images with any distribution against frequency analysis attacks.

B. RANDOMNESS ANALYSIS

However, to avoid information leaking, the encrypted image should be random on both the visual side and the statistics side. Therefore, a further statistical test is conducted using the SP 800-22 randomness test suit powered by NIST [53]. Meanwhile, the quality of the keys used in encryption is also tested. Then, after all the objects under test are transformed into a binary stream, their P-values (with a significance level $\alpha = 0.01$) are obtained and listed in Table 2. Since all the results are more than 0.01, the keys and cipher images are random in all aspects with a confidence of 99%. Hence, the proposed algorithm can resist attacks based on statistical analysis.

C. ANALYSIS OF CORRELATION BETWEEN ADJACENT PIXELS

Unlike text data, the strong inner correlation and high redundancy make it hard to completely hide the pattern of a plain image. One of the most important targets of image encryption is to minimize such inner correlation. The lower the correlation, the higher the security.

For this test, 10,000 pairs of adjacent pixels in all directions are randomly extracted from 'Baboon'. By comparing the scatter diagrams of the plain and cipher images illustrated in Figure 9, the highly concentrated distributions all become random distributions in any direction after 'Baboon' is encrypted by the proposed algorithm.

The indicator of γ_{xy} (correlation coefficient) provides an approach for quantitatively analyzing the degree of correlation between adjacent pixels, which defined as

$$\begin{cases} \bar{x} = \sum_{i=1}^{N} x_i/N, \bar{y} = \sum_{i=1}^{N} y_i/N \\ \gamma_{xy} = \dfrac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left(\sum_{i=1}^{N} (x_i - \bar{x})^2\right)\left(\sum_{i=1}^{N} (y_i - \bar{y})^2\right)}} \end{cases} \tag{9}$$

where x_i and y_i are two adjacent pixels in an image. Then, after N is set to10,000, the results are shown in Table 3.

Based on 'Baboon', Table 4 compares γ_{xy} of this work and other works. Therefore, the lower correlation coefficients between adjacent pixels of this work indicate better resistance against statistical attacks.

D. INFORMATION ENTROPY ANALYSIS

Information entropy is an essential measurement to judge the disorder level of images, and a higher entropy indicates a more random image. As given by Equation (10), the global entropy $H(g)$ can measure the global uncertainty of an image.

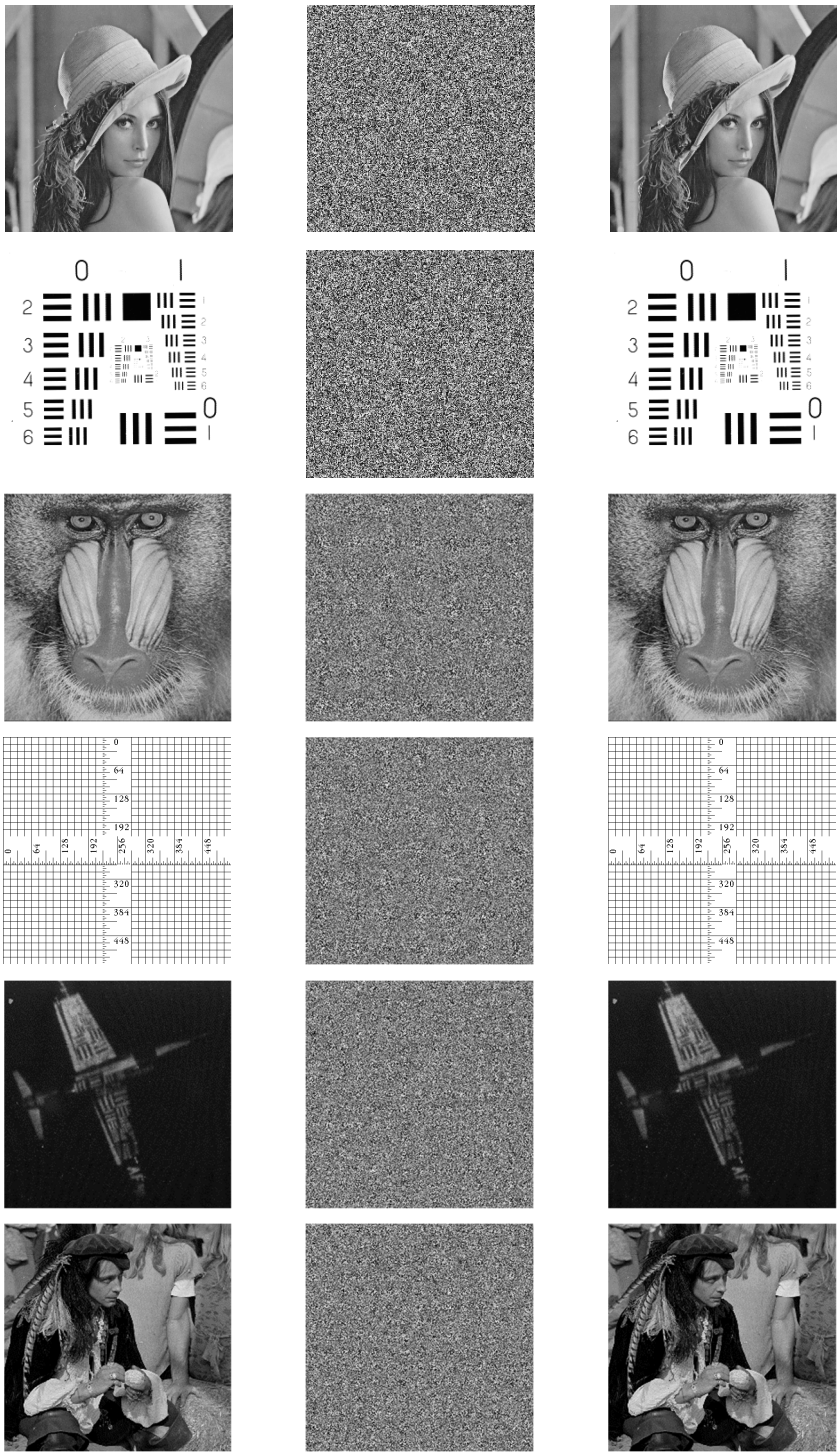


FIGURE 7. Results of encryption and decryption. Plain images locate in the first column, while their corresponding cipher images and decrypted images locate in the second and third columns, respectively.

Here, g_i represents a pixel value (for an 8-bit grayscale image, L is a integer between 0 and 255), and $p(g_i)$ expresses the proportion of each pixel value.

$$H(g) = -\sum_{i=0}^{L} p(g_i) \log_2 p(g_i) \tag{10}$$

In an ideal cipher image, all pixel values are equiprobable, indicating that the expected maximum $H(g) = 8$. Since the results of entropy tests listed in Table 5 are all very close to 8, it is confirmed that our algorithm achieves equivalent or even better performance in resisting global entropy analysis-based attacks.

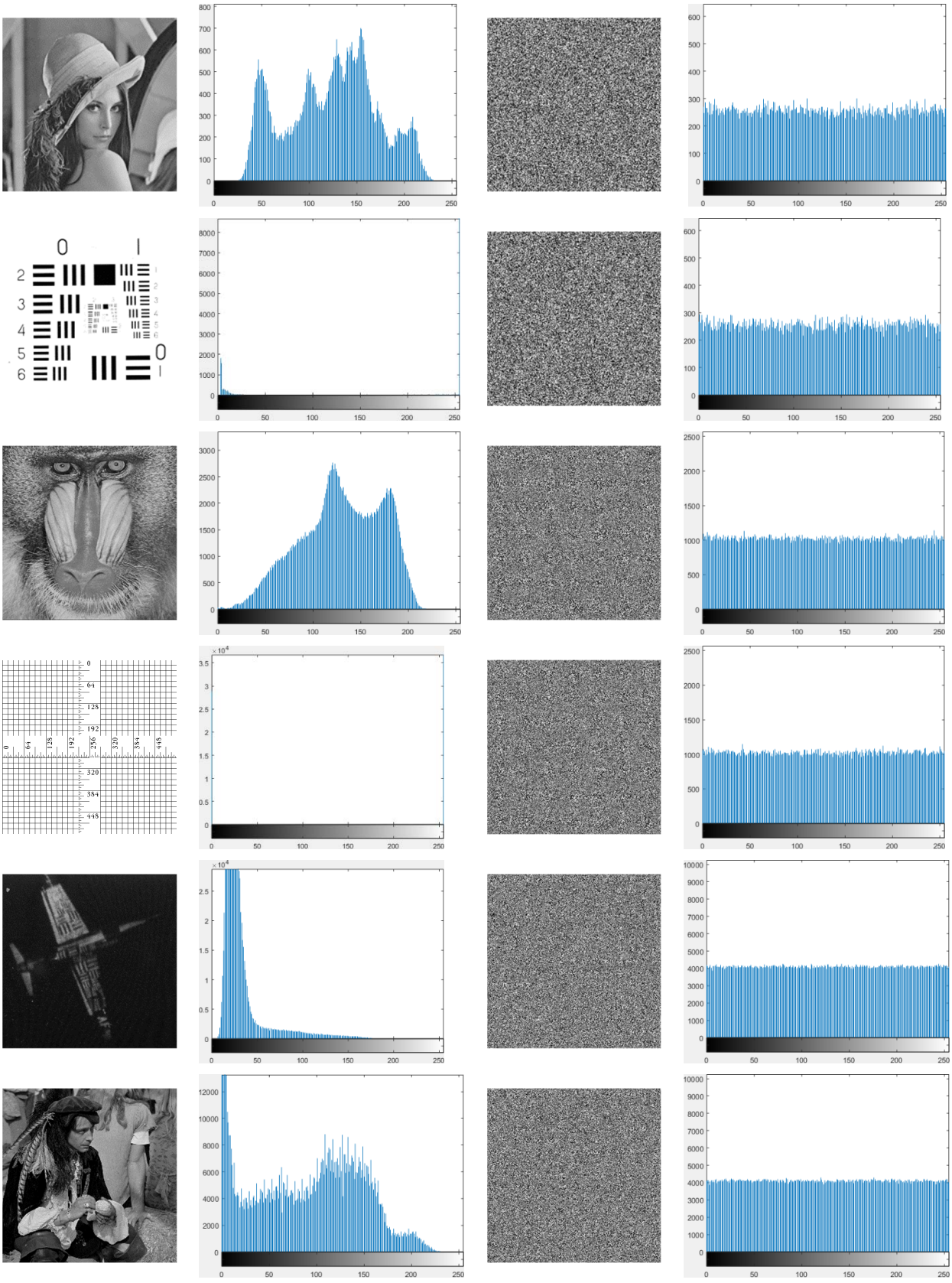


FIGURE 8. Results of histogram test. The second column presents histograms of plain images in column one. The fourth column presents histograms of cipher images in column three.

TABLE 1. Results of chi-square test.

Image	Theoretical value	χ^2 value		*P*-value		Flag
		Plain	Cipher	Plain	Cipher	
Lena	293.24738	3.9869×10^4	242.3578	0	0.9864	Pass
Resolution	293.24738	1.1198×10^4	276.3485	0	0.7530	Pass
Baboon	293.24738	1.8736×10^5	242.4456	0	0.9863	Pass
Ruler	293.24738	5.3703×10^7	257.9455	0	0.9323	Pass
Male	293.24738	7.0934×10^5	275.9512	0	0.7584	Pass
Airplane	293.24738	7.1999×10^6	233.2953	0	0.9959	Pass

TABLE 2. Results of randomness tests.

Test	*P*-value (set α=0.01)							
	$\boldsymbol{K}_\mathrm{d}$	$\boldsymbol{K}_\mathrm{p}$	Lena	Resolution	Baboon	Peppers	Airplane	Male
Frequency	0.1756	0.6088	0.665	0.6386	0.4164	0.9929	0.3453	0.4403
Block Frequency	0.5942	0.1098	0.0804	0.0831	0.6061	0.8314	0.6915	0.3581
Cumulative Sums	0.0605	0.2334	0.2073	0.2980	0.3474	0.0404	0.3704	0.1014
Runs	0.0345	0.2370	0.8916	0.1734	0.0122	0.1617	0.1843	0.5606
Longest Runs of Ones	0.2292	0.1166	0.9795	0.7188	0.7501	0.4774	0.2974	0.0368
Rank	0.0502	0.6181	0.2589	0.5442	0.0302	0.5019	0.9850	0.6889
Spectral DFT	0.1756	0.0310	0.9509	0.5158	0.9666	0.5906	0.3995	0.9923
Nonperiodic Template Matchings	0.0805	0.1177	0.5865	0.4990	0.6277	0.9802	0.5209	0.1411
Overlapping Template Matchings	0.6789	0.3073	0.3528	0.3014	0.3518	0.3282	0.1643	0.9812
Universal Statistical	0.7424	0.9041	0.4406	0.7851	0.7952	0.9225	0.1989	0.9672
Approximate Entropy	0.1778	0.8294	0.1266	0.9466	0.3588	0.7542	0.4902	0.1422
Random Excursions	0.1887	0.0486	0.3741	0.7874	0.6749	0.9971	0.5003	0.6714
Random Excursions Variant	0.9461	0.7099	0.8732	0.3425	0.1553	0.3931	0.9715	0.6639
Serial	0.2516	0.4588	0.0564	0.8977	0.4545	0.8344	0.8572	0.6604
Linear Complexity	0.0351	0.4587	0.7645	0.2095	0.9311	0.4752	0.7578	0.6458
Flag	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

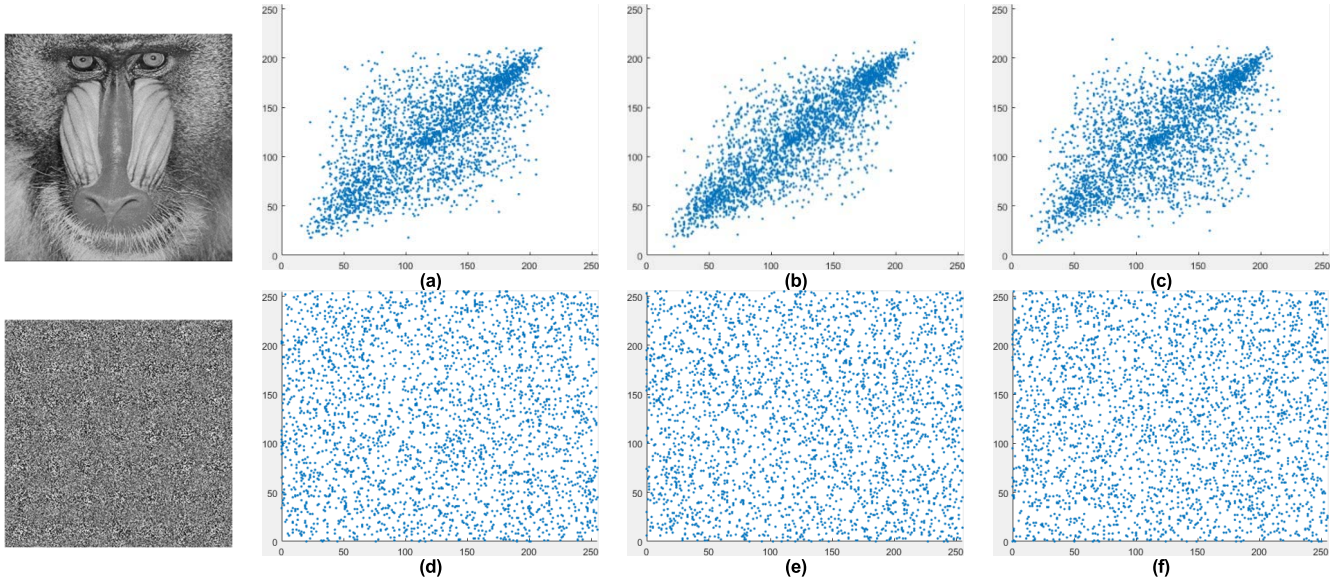


FIGURE 9. Scatter diagrams of the plain image in (a) the vertical direction, (b) the horizontal direction, and (c) the diagonal direction. The scatters of the cipher image in (d) the vertical direction, (e) the horizontal direction, and (f) the diagonal direction.

In 2013, a more rigorous test called local entropy was proposed [54], which could detect the limitations of global entropy. Its equation is expressed by

$$\bar{H}_{k,Tb}(S) = \sum_{i=1}^{k} \frac{H(S_i)}{k} \tag{11}$$

where S_i $(i = 1, 2, \ldots, S_k)$ are k non-overlapping sub-images containing T_b pixels randomly extracted from an image. The ideal value of the local entropy should be 7.902469317 (set $(k, T_b) = (30, 1936)$ and $\alpha = 0.05$), but generally, if the results fall into the range from 7.901901305 to 7.903037329, the tests could be considered passed. As listed in Table 6,

TABLE 3. Results of correlation coefficients.

Image	Vertical		Horizontal		Diagonal	
	Plain	Cipher	Plain	Cipher	Plain	Cipher
Lena	0.9575	-0.0019	0.9203	-0.0009	0.9148	0.0025
Resolution	0.8668	-0.0003	0.8722	-0.0027	0.7568	0.0011
Baboon	0.7524	0.0012	0.8653	-0.0004	0.7209	0.0004
Ruler	0.4599	-0.0021	0.4494	-0.0028	-0.0291	-0.0003
Male	0.9804	0.0002	0.9769	0.0001	0.9669	0.0006
Airplane	0.9458	0.0006	0.9647	-0.0008	0.9442	-0.0002

TABLE 4. Comparisons between different algorithms.

Algorithm	Vertical	Horizontal	Diagonal
Proposed	0.0012	-0.0004	0.0004
Ref. [40]	0.0230	0.0054	-0.0168
Ref. [41]	-0.0013	0.0033	-0.0009
Ref. [44]	-0.0017	0.0012	-0.0020
Ref. [45]	-0.0021	-0.0018	-0.0015
Ref. [49]	-0.0024	-0.0069	0.0007
Ref. [50]	0.0005	-0.0003	-0.0004

TABLE 5. Results of global entropy test.

Image	Plain	Cipher						
		Ours	Ref. [40]	Ref. [41]	Ref. [44]	Ref. [45]	Ref. [49]	Ref. [50]
Lena	7.4429	7.9973	7.9972	7.9991	7.9973	7.9992	7.9971	7.9914
Resolution	1.5483	7.9970	7.9968	7.9992	7.9972	7.9991	7.9967	7.9917
Baboon	7.3583	7.9992	7.9966	7.9992	7.9992	7.9993	7.9981	7.9921
Ruler	0.5000	7.9993	7.9971	7.9989	7.9993	7.9989	7.9968	7.9911
Male	5.6415	7.9998	7.9974	7.9990	7.9998	7.9993	7.9972	7.9916
Airplane	7.5237	7.9999	7.9968	7.9993	7.9998	7.9994	7.9975	7.9921

since all local entropies of cipher images fall into the acceptance interval, it is confirmed that the proposed image encryption algorithm provides enough security to resist strong attacks of local entropy analysis.

E. PLAINTEXT SENSITIVITY ANALYSIS

As mentioned earlier, a secure encryption algorithm should output two completely different cipher images when two plain images with a slight distinction are input. This guarantees the capability of a cryptosystem to resist strong differential attacks. In general, NPCR and UACI are two key indicators to represent the degree of difference between the two images. Suppose $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are two different cipher images, and $\boldsymbol{D}(i,j)=0$ if $\boldsymbol{C}_1(i,j)=\boldsymbol{C}_2(i,j)$ or $\boldsymbol{D}(i,j)=1$ if $\boldsymbol{C}_1(i,j)\neq\boldsymbol{C}_2(i,j)$. Then, NPCR and UACI can be defined as

$$\begin{cases} \text{NPCR} = \sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{D}(i,j)/(HW)\times 100\% \\ \text{UACI} = \sum_{i=1}^{H}\sum_{j=1}^{W}\frac{|\boldsymbol{C}_1(i,j)-\boldsymbol{C}_2(i,j)|}{255}/(HW)\times 100\% \end{cases} \tag{12}$$

For strict evaluation, this paper randomly picks one pixel from a test image, modifies its least significant digit, and then calculates NPCR and UACI between encryption results. After the above test is repeated 100 times, the average values are listed in Table 7. By comparison with the expected values of other works, it is confirmed that the proposed algorithm can provide enough security even facing strong differential attacks.

F. KEY SENSITIVITY ANALYSIS

In practical applications, all details of an encryption algorithm are overt except the session key. This indicates that the privacy protection of image data depends on the quality of the session key. For this purpose, the cipher image must be highly sensitive to even slight modifications of the session key in two aspects: 1) Tiny changes of the session key will lead to a significant difference between outputs when encrypting the same plain image; 2) Subtle perturbations of the session key will result in a complete failure of decryption.

Using 'Ruler' as the test image, its corresponding session key $\boldsymbol{SK}$ = {5D64E5EC976C6C830C2555BCBE9B03294DC5A94B8346579082C2B73EF5AD1643} is obtained via *Step A-2*. Then, following *Step A-3* and *Step A-4*, $\boldsymbol{SK}$ will be split into nine parts to extract the initial conditions of chaotic systems.

In phase one, the least significant digit of nine parts is successively modified, and NPCR and UACI are calculated

TABLE 6. Results of local entropy test.

Image	Lena	Resolution	Baboon	Ruler	Male	Airplane
Local entropy	7.902254141	7.902595240	7.902102711	7.902754893	7.901978147	7.902551637
Flag	Pass	Pass	Pass	Pass	Pass	Pass

TABLE 7. Average values of NPCR and UACI test (Unit: %).

Image	NPCR					UACI				
	Expected	Ours	Ref. [40]	Ref. [44]	Ref. [45]	Expected	Ours	Ref. [40]	Ref. [44]	Ref. [45]
Lena	99.6094	99.6088	99.6216	99.61	99.7955	33.4635	33.4667	33.4994	33.49	34.0486
Resolution		99.6096	99.6307	99.61	99.6915		33.4609	33.4751	33.49	33.6281
Baboon		99.6098	99.6368	99.61	99.6669		33.4642	33.4702	33.48	33.6207
Ruler		99.6097	99.5936	99.61	99.6723		33.4635	33.4829	33.47	33.6125
Male		99.6096	99.5773	99.61	99.6931		33.4634	33.5008	33.47	33.6473
Airplane		99.6102	99.6227	99.61	99.5899		33.4629	33.4761	33.48	34.0028

between cipher images. As shown in Table 8, all values of NPCR and UACI are close to 99.6094% and 33.4635%, indicating that minor changes in the session key can significantly affect the cipher image when encryption. Thus, the proposed algorithm is highly sensitive to minor modifications of the encryption key.

In phase two, the correct session key and modified session key are used to decrypt the cipher image, respectively. As illustrated in Figure 10, only the correct session key can recover the plain image while other decrypted images are still disordered.

Then, NPCR and UACI are used to further evaluate the degree of the difference between the cipher image and the above decrypted images. The results listed in Table 9 show that tiny errors in the session key will cause the failure of decryption without leaking any useful information. Therefore, our algorithm also has a high sensitivity to the decryption key.

G. KEY SPACE ANALYSIS

Besides the high sensitivity to key changes, a large enough key space of the session key is also essential to enabling an encryption algorithm to resist brute-force attacks. In this paper, the 256-bit session key is obtained from the mixture of a true random sequence and a unique identification of a plain image, which ensures the one-time pad and randomness. In the above section, it has been proven that all bits of $\boldsymbol{SK}$ are valid to affect the output of encryption and decryption. Then, our algorithm can provide 2^{256} different session keys, which is much bigger than a secure threshold of 2^{128} to resist key exhaustion attacks [40], [55].

H. CHOSEN-PLAINTEXT ATTACKS ANALYSIS

The CPA (chosen-plaintext attack) is always a top threat to an applicable encryption algorithm. Commonly, 'Black' (with all pixels being 0) and 'White' (with all pixels being 255) are two often used images to implement CPA. Hence, this paper performs all the above security tests to verify whether the proposed algorithm could resist CPA.

Figure 11 and Figure 12 show the results of histogram tests and scatter tests, while the other experimental results are listed in Table 10. It can be seen from Figure 11 and Figure 12 that the 'Black' image and 'White' image are successfully encrypted to random images at the visual level. Then, as illustrated in Table 10, all the other results meet the requirements of passing tests, which further indicates that our algorithm can resist CPA.

I. TIME COMPLEXITY ANALYSIS

The previous sections have confirmed that the proposed image encryption algorithm has high security to resist common attacks. However, for real-time communication applications, fast encryption speed is also indispensable. Hence, the time complexity of the proposed algorithm and its actual execution time on the Matlab platform are analyzed. Since the session key could be obtained before encrypting images, and the time consumption of padding or removing padding of images is quite negligible, this paper only discusses the time consumption of other parts of the proposed image encryption algorithm.

For the generation method of diffusion keys, referring to **Algorithm 2**, its time complexity is $\boldsymbol{O}(WH/2)$. The generation method of permutation keys, referring to *Step D-1* to *Step D-5*, has a time complexity of $\boldsymbol{O}(2W + 2H)$. The generation method of initial vector, referring to *Step E-1* to *Step E-5*, has a complexity of $\boldsymbol{O}(W)$. The encryption process, referring to **Algorithm 3**, has a time complexity of $\boldsymbol{O}(W+H)$. Then, our algorithm realizes a time complexity of $\boldsymbol{O}(WH/2+ 4W + 3H)$. Attributed to the use of the novel divide and conquer diffusion strategy, the efficiency of our algorithm is superior to that of other CBC-based works.

Then, the actual execution time of the proposed algorithm is listed in Table 11. Even considering the low efficiency of MATLAB, our algorithm still achieves satisfactory results in comparison with other works. This is attributed to the optimization of the encryption process and the faster iteration speed of 2D hyperchaotic systems than high-order continuous hyperchaotic systems. Combined with the previous

TABLE 8. Results of the encryption key sensitivity test (Unit: %).

Session key	Changed value	NPCR	UACI
5D64E5EC976C	9C5C00CFCFBD	99.6162	33.4093
76C6C830C255	420A453AF7F4	99.6261	33.4578
2555BCBE9B03	5043C1121B42	99.6029	33.4579
B03294DC5A94	ACE25F0ED6B5	99.6135	33.4063
A94B834657 90	7EAD7CC964A1	99.6037	33.4393
79082C2B73EF	585A5F10476E	99.6231	33.4966
5AD	E09FD648479C	99.6120	33.4106
D16	12FEB7	99.6022	33.4124
643	19AFE2	99.6136	33.4932

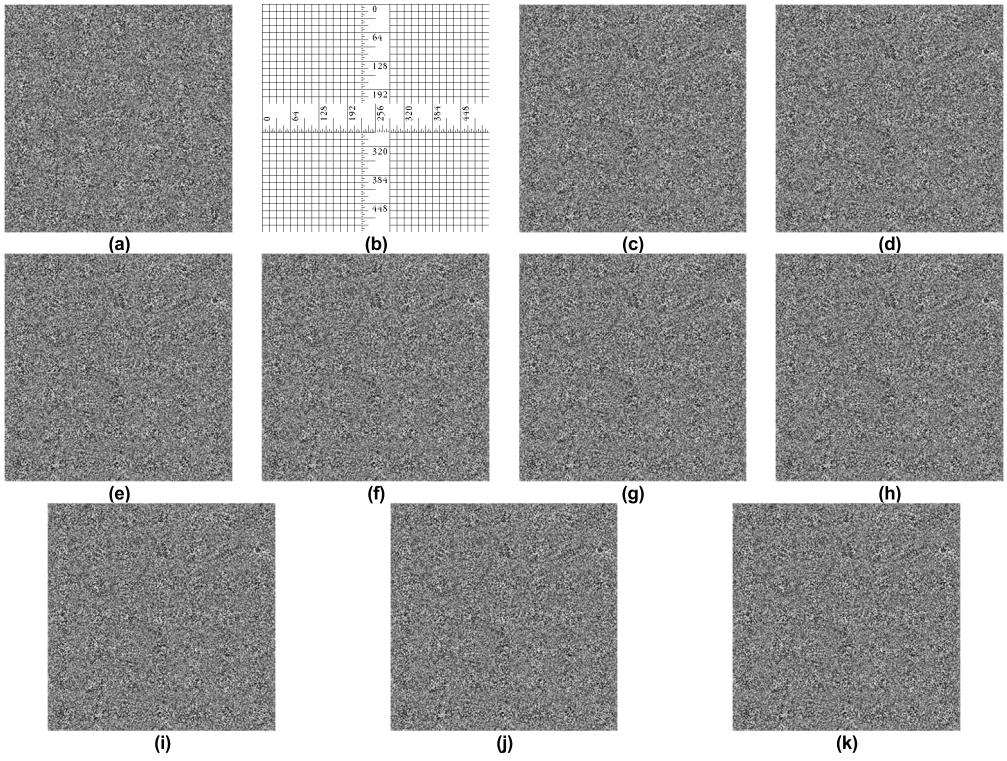


FIGURE 10. Decryption results from (a) the encrypted 'Ruler' with (b) a correct session key, and an incorrect session key by changing (c) $SK(12) = D$, (d) $SK(21) = 4$, (e) $SK(30) = 2$, (f) $SK(39) = 5$, (g) $SK(48) = 1$, (h) $SK(57) = E$, (i) $SK(60) = C$, (j) $SK(62) = 7$, (k) $SK(64) = 2$.

TABLE 9. Results of the decryption key sensitivity test (Unit: %).

Session key	Modified value	NPCR	UACI
5D64E5EC976C	9C5C00CFCFBD	99.6041	33.5040
76C6C830C255	420A453AF7F4	99.5946	33.4495
2555BCBE9B03	5043C1121B42	99.6079	33.4204
B03294DC5A94	ACE25F0ED6B5	99.6310	33.4419
A94B834657 90	7EAD7CC964A1	99.6105	33.4717
79082C2B73EF	585A5F10476E	99.6048	33.4766
5AD	E09FD648479C	99.6318	33.4417
D16	12FEB7	99.6098	33.4488
643	19AFE2	99.6037	33.4601

outstanding performance of security tests, it can be concluded that the proposed algorithm is suitable for massive image data real-time communications.

J. ROBUSTNESS ANALYSIS

During data transmission, data loss and noise pollution are often unavoidable. However, due to its high redundancy,

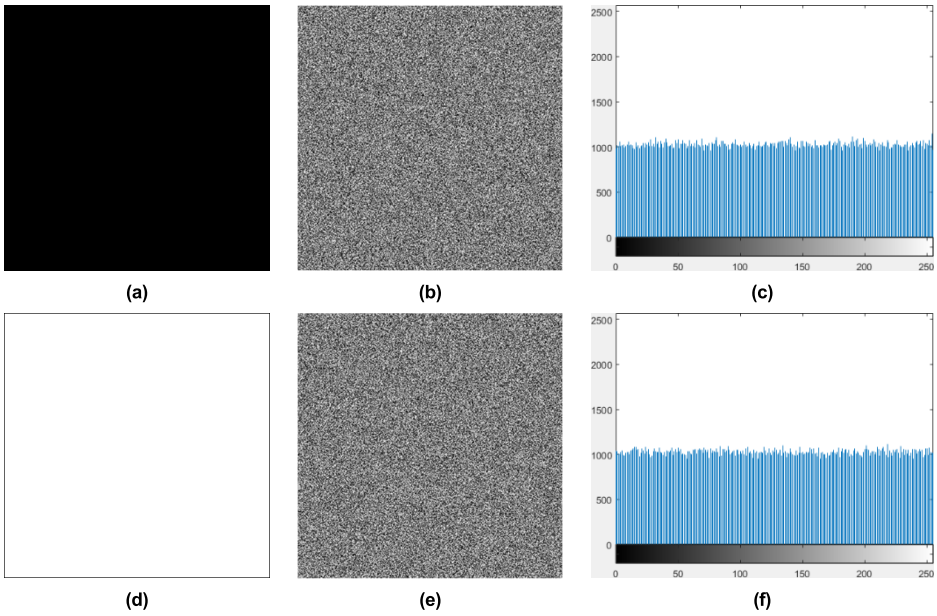


FIGURE 11. (a) 'Black' image, (b) its cipher image, and (c) the histogram of (b). (d) 'White' image, (e) its cipher image, and (f) the histogram of (e).

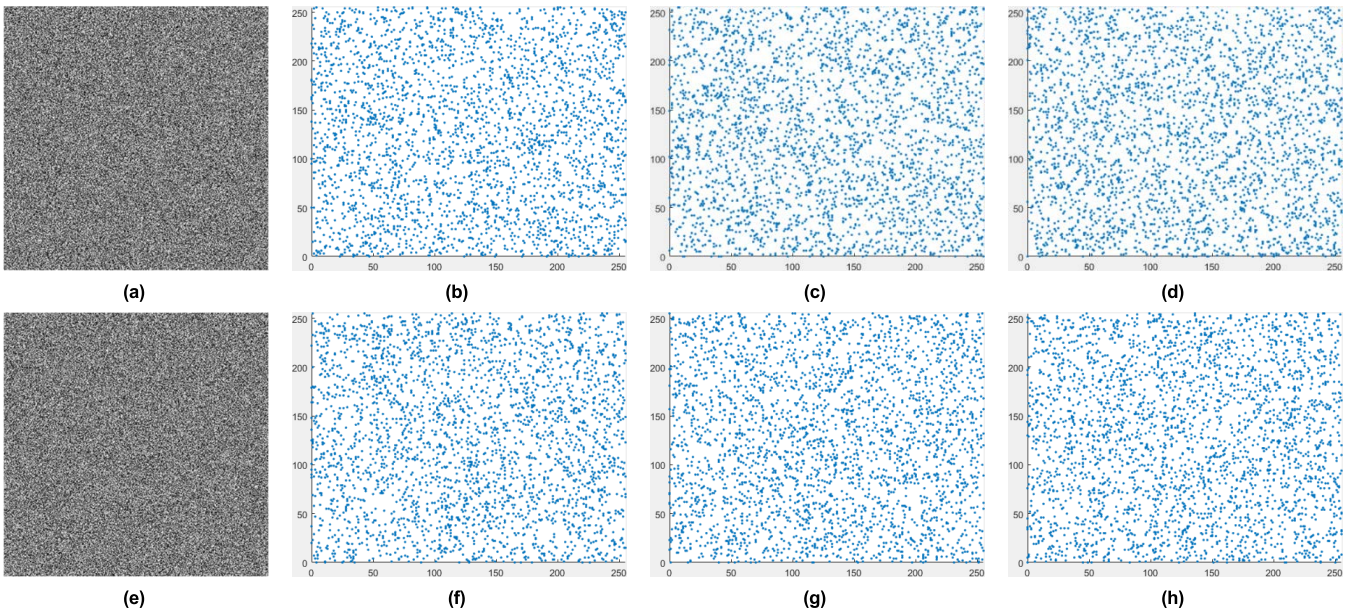


FIGURE 12. Scatter diagrams of (a) the encrypted 'Black' image and (e) 'White' image in the vertical direction, horizontal direction, and diagonal direction.

TABLE 10. Security performance of our algorithm on 'Black' image and 'White' image.

Test	χ^2 test		NPCR (%)		UACI (%)		Entropy		γ_{xy}		
	χ^2 value	P-value	Min	Max	Min	Max	Global	Local	Horizontal	Vertical	Diagonal
Black	250.4767	0.9665	99.6107	99.6327	33.4388	33.4713	7.9993	7.9025	0.0023	-0.0006	-0.0011
White	254.5701	0.9501	99.6219	99.6337	33.4450	33.4689	7.9994	7.9023	0.0014	-0.0011	0.0013

minor errors do not affect the overall intelligibility of image data. Hence, a useful image cryptosystem should have the capability to recover a cipher image polluted by noise or encountering data loss.

Figure 13 demonstrates the decryption results of cipher images with data loss of varying degrees during transmission, which indicates that even though half the pixels are discarded, our algorithm can still recover it to a readable image.

TABLE 11. Results of execution times (Unit: s).

Image	Size	Ours	Ref. [40]	Ref. [41]	Ref. [44]	Ref. [45]	Ref. [49]	Ref. [50]
Lena	256×256	0.0198	0.0899	0.1837	0.1623	0.0702	0.1306	0.4598
Resolution		0.0205	0.0815	0.1674	0.1591	0.0617	0.1299	0.4367
Baboon	512×512	0.0799	0.2964	0.7075	0.3645	0.3469	0.5407	1.7697
Ruler		0.0782	0.2833	0.6978	0.3479	0.3701	0.5023	1.6635
Male	1024×1024	0.3139	1.0235	3.0374	1.0145	1.6331	2.6773	7.9187
Airplane		0.3107	0.9837	2.9281	1.1078	1.5819	2.5981	7.8811

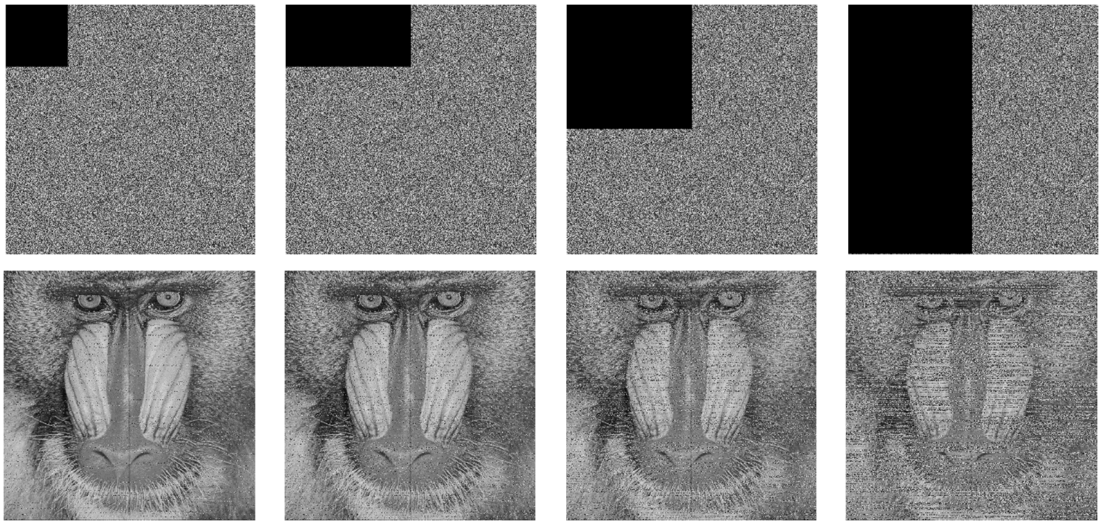


FIGURE 13. Decryption results of cipher images with data lost (a) 1/16, (b) 1/8, (c) 1/4, and (d) 1/2.

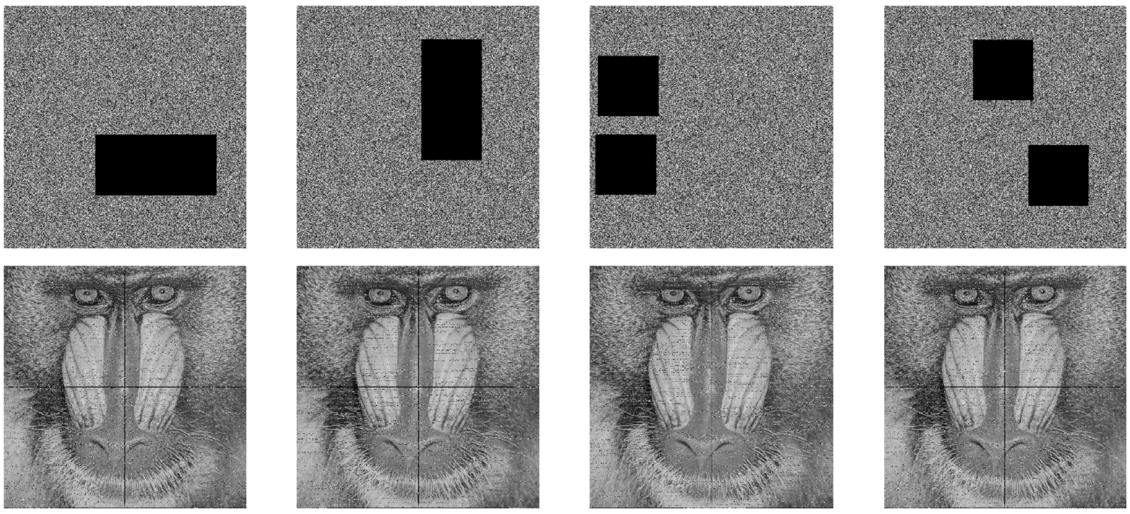


FIGURE 14. Decryption results of cipher images with data lost 1/8 in different locations.

Figure 14 shows the decrypted images with a data loss of 1/8 in different locations during transmission, meaning the location of data loss does not affect the decryption. Hence, our algorithm has strong robustness to data loss in complex open channels.

Figure 15 shows the decryption results of cipher images polluted by common noise of different intensities during transmission. Since some important information can still be recognized to confirm 'Baboon' images, the proposed algorithm has certain robustness to resist noise pollution in low-quality channels.

Moreover, this paper utilizes PSNR test to check the quality of the above decrypted images, which defined as

$$\mathrm{PSNR} = 10\lg \frac{255 \times 255}{\sum_{i=1}^{W}\sum_{j=1}^{H} (\boldsymbol{R}(i,j) - \boldsymbol{D}(i,j))^2/(WH)} \tag{13}$$

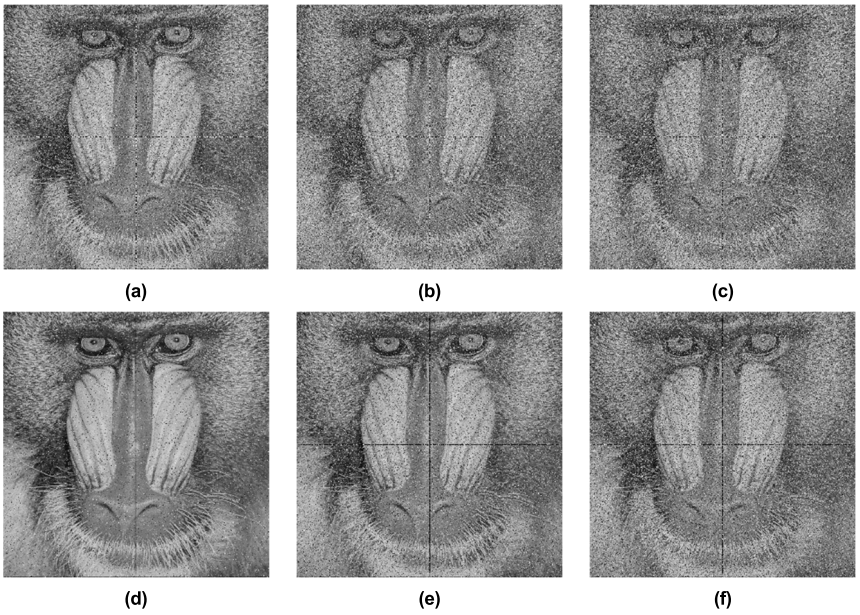


FIGURE 15. Decryption results of cipher images polluted by Gaussian noise with an intensity of (a) 0.001, (b) 0.003, (c) 0.005, and polluted by Salt&Peppers noise with an intensity of (d) 0.01, (e) 0.03, (f) 0.05.

TABLE 12. Results of PSNR test.

Type	Data lost				Polluted by Gaussian			Polluted by Salt&Peppers		
Intensity	1/16	1/8	1/4	1/2	0.001	0.003	0.005	0.01	0.03	0.05
PSNR	21.4577	18.3846	15.4323	12.5127	15.4017	13.4280	12.7872	20.6370	16.2307	14.3321

where $\boldsymbol{R}$ represents the reference image and $\boldsymbol{D}$ represents the test image with data loss or pollution. As demonstrated in Table 12, for the above decrypted images, all results of the PSNR test exceed 10dB, which further proves the proposed algorithm can provide privacy protection properly even on an unstable channel.

V. CONCLUSION

This paper presents a fast chaotic image encryption algorithm using a novel divide and conquer diffusion strategy. First, for any plain image, a 256-bit one-time session key is obtained from a mixture of its hash values and a true random sequence. Second, a pseudo-random sequence generator is designed to produce diffusion keys. The employed 2D hyperchaotic systems, CTDHC and GTDHCS, overcome the high time consumption of solving continuous hyperchaotic systems. In addition, the bitwise XOR operation between quantified hyperchaotic sequences are performed to further enhance the statistical properties and anti-attack performance of the final key stream for diffusion. Then, we present a novel divide and conquer diffusion strategy to accelerate the CBC-based encryption scheme. In phase one, the plain image is divided into three parts, thus a simultaneous diffusion can work on upper and bottom parts. In particular, the encryption is launched by an initial vector provided by CLSCM. Then, to implement sufficient diffusion in vertical direction, we must repeat the previous diffusion process by acting the central row (which is already influenced by both upper and bottom parts) as a new initial vector. Next, the temporary cipher image will be divided into three parts (left part, right part, and central column), thus a similar diffusion process as phase one but in horizontal direction (here, it is launched by the central column of the temporary cipher image) will be performed to achieve full encryption, which guarantees that any minor changes in the plain image can lead to significant changes in the cipher image. Moreover, within the diffusion process, CLLCM is employed to randomly scramble pixel positions to further enhance the security of the proposed algorithm. In addition, to make our algorithm adapt to images of any size, a padding strategy and its corresponding removing strategy are also presented. Experimental results and comparisons indicate that the proposed algorithm can provide more outstanding security performance to resist common attacks such as frequency analysis, differential analysis, and statistical analysis because its P-values of χ^2 tests are much larger than 0.05, P-values of NIST SP-800 22 tests are much larger than 0.01, correlation coefficients are closer to 0, global entropies are closer to 8, and values of NPCR and UACI tests are approaching ideal values. More importantly, due to the time complexity of $\boldsymbol{O}(W + H)$ of our encryption process, our algorithm only costs 0.02s, 0.08s, and 0.31s to encrypt images of sizes 256×256, 512×512, and

1024×1024, which meets the high efficiency requirements of real-time transmission. Overall, with continuous research in chaos theory and parallel techniques, our algorithm can provide security solutions for image data communication scenarios.

REFERENCES

[1] 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, doi: 10.1371/journal.pone.0260014.

[2] N. Khalil, A. Sarhan, and M. A. M. Alshewimy, "An efficient color/grayscale image encryption scheme based on hybrid chaotic maps," *Opt. Laser Technol.*, vol. 143, Nov. 2021, Art. no. 107326, doi: 10.1016/j.optlastec.2021.107326.

[3] X. Wang, Y. Su, M. Xu, H. Zhang, and Y. Zhang, "A new image encryption algorithm based on Latin square matrix," *Nonlinear Dyn.*, vol. 107, no. 1, pp. 1277–1293, 2022, doi: 10.1007/s11071-021-07017-7.

[4] N. Munir, M. Khan, A. Al Karim Haj Ismail, and I. Hussain, "Cryptanalysis and improvement of novel image encryption technique using hybrid method of discrete dynamical chaotic maps and Brownian motion," *Multimedia Tools Appl.*, vol. 81, no. 5, pp. 6571–6584, 2022, doi: 10.1007/s11042-021-11810-2.

[5] J. M. K. Mastan and R. Pandian, "Cryptanalysis of two similar chaos-based image encryption schemes," *Cryptologia*, vol. 45, no. 6, pp. 541–552, Nov. 2021, doi: 10.1080/01611194.2020.1814447.

[6] M. Shen, Y. Deng, L. Zhu, X. Du, and N. Guizani, "Privacy-preserving image retrieval for medical IoT systems: A blockchain-based approach," *IEEE Netw.*, vol. 33, no. 5, pp. 27–33, Sep. 2019, doi: 10.1109/MNET.001.1800503.

[7] Z. Kuang, Z. Guo, J. Fang, J. Yu, N. Babaguchi, and J. Fan, "Unnoticeable synthetic face replacement for image privacy protection," *Neurocomputing*, vol. 457, pp. 322–333, 2021, doi: 10.1016/j.neucom.2021.06.061.

[8] X. Chai, Y. Wang, Z. Gan, X. Chen, and Y. Zhang, "Preserving privacy while revealing thumbnail for content-based encrypted image retrieval in the cloud," *Inf. Sci.*, vol. 604, pp. 115–141, Aug. 2022, doi: 10.1016/j.ins.2022.05.008.

[9] *Data Encryption Standard (DES)*, FIPS Standard 46-3, 1999.

[10] *Advanced Encryption Standard (AES)*, FIPS Standard, 197, 2001.

[11] S. 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, doi: 10.4103/0256-4602.72583.

[12] B. Zhang, D. Xiao, and Y. Xiang, "Robust coding of encrypted images via 2D compressed sensing," *IEEE Trans. Multimedia*, vol. 23, pp. 2656–2671, 2021, doi: 10.1109/TMM.2020.3014489.

[13] X. Wang and Y. Su, "Image encryption based on compressed sensing and DNA encoding," *Signal Process., Image Commun.*, vol. 95, Jul. 2021, Art. no. 116246, doi: 10.1016/j.image.2021.116246.

[14] C. Zou, X. Wang, and H. Li, "Image encryption algorithm with matrix semi-tensor product," *Nonlinear Dyn.*, vol. 105, no. 1, pp. 859–876, 2021, doi: 10.1007/s11071-021-06542-9.

[15] C. Wang and Y. Zhang, "A novel image encryption algorithm with deep neural network," *Signal Process.*, vol. 196, Jul. 2022, Art. no. 108536, doi: 10.1016/j.sigpro.2022.108536.

[16] X. Zhang and X. Yan, "Adaptive chaotic image encryption algorithm based on RNA and pixel depth," *Electronics*, vol. 10, no. 15, p. 1770, Jul. 2021, doi: 10.3390/electronics10151770.

[17] X. Zhang and Z. Gong, "Color image encryption algorithm based on 3D zigzag transformation and view planes," *Multimedia Tools Appl.*, vol. 81, pp. 31753–31785, Apr. 2022, doi: 10.1007/s11042-022-13003-x.

[18] N. Rani, S. R. Sharma, and V. Mishra, "Grayscale and colored image encryption model using a novel fused magic cube," *Nonlinear Dyn.*, vol. 108, no. 2, pp. 1773–1796, 2022, doi: 10.1007/s11071-022-07276-y.

[19] Y. Naseer, T. Shah, and D. Shah, "A novel hybrid permutation substitution base colored image encryption scheme for multimedia data," *J. Inf. Secur. Appl.*, vol. 59, Jul. 2021, Art. no. 102829, doi: 10.1016/j.jisa.2021.102829.

[20] L. Qu, F. Chen, S. Zhang, and H. He, "Cryptanalysis of reversible data hiding in encrypted images by block permutation and co-modulation," *IEEE Trans. Multimedia*, vol. 24, pp. 2927–2937, 2021, doi: 10.1109/TMM.2021.3090588.

[21] H. Li, T. Li, W. Feng, J. Zhang, J. Zhang, L. Gan, and C. Li, "A novel image encryption scheme based on non-adjacent parallelable permutation and dynamic DNA-level two-way diffusion," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102844, doi: 10.1016/j.jisa.2021.102844.

[22] X. Chai, Y. Tian, Z. Gan, Y. Lu, X. J. Wu, and G. Long, "A robust compressed sensing image encryption algorithm based on GAN and CNN," *J. Mod. Opt.*, vol. 69, no. 2, pp. 103–120, 2022, doi: 10.1080/09500340.2021.2002450.

[23] 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, doi: 10.1109/ACCESS.2019.2959137.

[24] S. Zhou, Z. Zhao, and X. Wang, "Novel chaotic colour image cryptosystem with deep learning," *Chaos, Solitons Fractals*, vol. 161, Aug. 2022, Art. no. 112380, doi: 10.1016/j.chaos.2022.112380.

[25] I. Koyuncu, M. Tuna, I. Pehlivan, C. B. Fidan, and M. Alçin, "Design, FPGA implementation and statistical analysis of chaos-ring based dual entropy core true random number generator," *Analog Integr. Circuits Signal Process.*, vol. 102, no. 2, pp. 445–456, 2020, doi: 10.1007/s10470-019-01568-x.

[26] A. V. Tutueva, T. I. Karimov, L. Moysis, E. G. Nepomuceno, C. Volos, and D. N. Butusov, "Improving chaos-based pseudo-random generators in finite-precision arithmetic," *Nonlinear Dyn.*, vol. 104, no. 1, pp. 727–737, 2021, doi: 10.1007/s11071-021-06246-0.

[27] T. L. Liao, P. Y. Wan, and J.-J. Yan, "Design and synchronization of chaos-based true random number generators and its FPGA implementation," *IEEE Access*, vol. 10, pp. 8279–8286, 2022, doi: 10.1109/ACCESS.2022.3142536.

[28] S. Zhou, X. Wang, W. Zhou, and C. Zhang, "Recognition of the scale-free interval for calculating the correlation dimension using machine learning from chaotic time series," *Phys. A, Stat. Mech. Appl.*, vol. 588, Feb. 2022, Art. no. 126563, doi: 10.1016/j.physa.2021.126563.

[29] R. Matthews, "On the derivation of a, 'chaotic' encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, 1989, doi: 10.1080/0161-118991863745.

[30] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, no. 8, pp. 821–824, 1990, doi: 10.1063/1.4917383.

[31] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998, doi: 10.1142/S021812749800098X.

[32] W. Feng, Y.-G. He, H.-M. Li, and C.-L. Li, "Cryptanalysis of the integrated chaotic systems based image encryption algorithm," *Optik*, vol. 186, pp. 449–457, Jun. 2019, doi: 10.1016/j.ijleo.2018.12.103.

[33] W. Feng and J. Zhang, "Cryptanalzing a novel hyper-chaotic image encryption scheme based on pixel-level filtering and DNA-level diffusion," *IEEE Access*, vol. 8, pp. 209471–209482, 2020, doi: 10.1109/ACCESS.2020.3038006.

[34] 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, doi: 10.1109/ACCESS.2021.3123571.

[35] S. Liu, C. Guo, and J. T. Sheridan, "A review of optical image encryption techniques," *Opt. Laser Technol.*, vol. 57, pp. 327–342, Apr. 2014, doi: 10.1016/j.optlastec.2013.05.023.

[36] C. L. Li, Y. Zhou, H. M. Li, W. Feng, and J. R. Du, "Image encryption scheme with bit-level scrambling and multiplication diffusion," *Multimedia Tools Appl.*, vol. 80, no. 12, pp. 18479–18501, 2021, doi: 10.1007/s11042-021-10631-7.

[37] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, pp. 749–761, Jul. 2004, doi: 10.1016/j.chaos.2003.12.022.

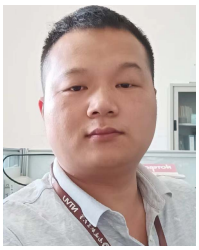
[38] Y. Mao, G. Chen, and S. Lian, "A novel fast image encryption scheme based on 3D chaotic baker maps," *Int. J. Bifurcation Chaos*, vol. 14, no. 10, pp. 3613–3624, Oct. 2004, doi: 10.1142/S021812740401151X.

[39] J. Chen, L. Yu Zhang, and Y. Zhou, "Re-evaluation of the security of a family of image diffusion mechanisms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4747–4758, Dec. 2021, doi: 10.1109/TCSVT.2021.3054508.

[40] L. Liu, Y. Lei, and D. Wang, "A fast chaotic image encryption scheme with simultaneous permutation-diffusion operation," *IEEE Access*, vol. 8, pp. 27361–27374, 2020, doi: 10.1109/ACCESS.2020.2971759.

[41] T. Li, J. Shi, and D. Zhang, "Color image encryption based on joint permutation and diffusion," *Proc. SPIE*, vol. 30, no. 1, 2021, Art. no. 013008, doi: 10.1117/1.JEI.30.1.013008.
[42] C. Fu, J.-J. Chen, H. Zou, W.-H. Meng, Y.-F. Zhan, and Y.-W. Yu, "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Opt. Exp.*, vol. 20, no. 3, pp. 2363–2378, 2012, doi: 10.1364/OE.20.002363.
[43] X. Zhang and Z. Zhao, "Chaos-based image encryption with total shuffling and bidirectional diffusion," *Nonlinear Dyn.*, vol. 75, nos. 1–2, pp. 319–330, Jan. 2014, doi: 10.1007/s11071-013-1068-4.
[44] H. X. Zhao, S. C. Xie, J. Z. Zhang, and T. Wu, "Efficient image encryption using two-dimensional enhanced hyperchaotic Henon map," *Proc. SPIE*, vol. 29, no. 2, 2020, Art. no. 023007, doi: 10.1117/1.JEI.29.2.023007.
[45] X. Wang and H. Zhao, "Fast image encryption algorithm based on parallel permutation-and-diffusion strategy," *Multimedia Tools Appl.*, vol. 79, no. 27, pp. 19005–19024, 2020, doi: 10.1007/s11042-020-08810-z.
[46] W. Feng, X. Zhao, J. Zhang, Z. Qin, J. Zhang, and Y. He, "Image encryption algorithm based on plane-level image filtering and discrete logarithmic transform," *Mathematics*, vol. 10, no. 15, p. 2751, 2022, doi: 10.3390/math10152751.
[47] K. Qian, W. Feng, Z. Qin, J. Zhang, X. Luo, and Z. Zhu, "A novel image encryption scheme based on memristive chaotic system and combining bidirectional bit-level cyclic shift and dynamic DNA-level diffusion," *Frontiers Phys.*, vol. 10, p. 718, Aug. 2022, doi: 10.3389/fphy.2022.963795.
[48] S. Zhou, X. Wang, Y. Zhang, B. Ge, M. Wang, and S. Gao, "A novel image encryption cryptosystem based on true random numbers and chaotic systems," *Multimedia Syst.*, vol. 28, pp. 95–112, May 2021, doi: 10.1007/s00530-021-00803-8.
[49] L. Teng, X. Wang, F. Yang, and Y. Xian, "Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion," *Nonlinear Dyn.*, vol. 105, no. 2, pp. 1859–1876, 2021, doi: 10.1016/j.optlastec.2021.107252.
[50] X. Gao, "Image encryption algorithm based on 2D hyperchaotic map," *Optics Laser Technol.*, vol. 142, Oct. 2021, Art. no. 107252, doi: 10.1016/j.optlastec.2021.107252.
[51] F. Yuan, Y. Li, and G. Wang, "A universal method of chaos cascade and its applications," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 31, no. 2, 2021, Art. no. 021102, doi: 10.1063/5.0041518.
[52] M. Haahr. *RANDOM.ORG: True Random Number Services*. Accessed: Jun. 28, 2022. [Online]. Available: https://www.random.org
[53] F. Pareschi, R. Rovatti, and G. Setti, "On statistical tests for randomness included in the NIST SP800–22 test suite and based on the binomial distribution," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 491–505, Apr. 2012, doi: 10.1109/TIFS.2012.2185227.
[54] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Inf. Sci.*, vol. 222, pp. 323–342, Feb. 2013, doi: 10.1016/j.ins.2012.07.049.
[55] Z. Li, C. Peng, L. Li, and X. Zhu, "A novel plaintext-related image encryption scheme using hyper-chaotic system," *Nonlinear Dyn.*, vol. 94, no. 2, pp. 1319–1333, 2018, doi: 10.1007/s11071-018-4426-4.

BIN GE received the Ph.D. degree in engineering from the University of Chinese Academy of Sciences, in 2016. He is currently a Lecturer with Nantong Vocational University. His current research interests include the theory and application of cryptography and chaotic cryptography.



ZHIHUA SHEN received the Ph.D. degree in engineering from Xi'an Jiao Tong University, in 2017. He is currently a Lecturer with Nantong Vocational University. His current research interests include chaos theory and circuit design.

JINBAO ZHANG received the Ph.D. degree in communication and information system from the Nanjing University of Aeronautics and Astronautics, in 2019. He is currently a Lecturer with Nantong University. His current research interests include physical security issues, side-channel analysis for cryptographic hardware and embedded systems, and application-specified integrated circuit design.