IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 10, 2018, accepted October 23, 2018, date of publication November 5, 2018, date of current version December 3, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2879565*

# A Chaos Based Image Encryption Scheme Using Digit-Level Permutation and Block Diffusion

**PING PING[1], JINYANG FAN[1], YINGCHI MAO[1], FENG XU[1], AND JERRY GAO[2]**
[1]College of Computer and Information, Hohai University, Nanjing 210098, China
[2]College of Engineering, San Jose State University, San Jose, CA 95192, USA

Corresponding author: Ping Ping (pingpingnjust@163.com)

This work was supported in part by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant 2016YFC0400910 and Grant 2017ZX07104001 and in part by the Fundamental Research Funds for the Central Universities under Grant 2017B16814 and Grant 2017B20914.

**ABSTRACT** Image encryption can be classified into pixel-level and bit-level according to the smallest processing element in the permutation and diffusion. Most pixel-level permutations only change the pixel positions, so the histograms of the permuted image and the original image are identical. The bit-level permutation can change the histogram of the image, but it is comparatively time-consuming due to bit-level computing. In this paper, we propose a new digit-level permutation. The pixel matrix of an image is decomposed into three digital matrices, and then these digital matrices are shuffled by the Henon map. This digit-level permutation combines the merits of pixel-level permutation with that of bit-level permutation. Besides, we also design a high-speed diffusion operation which exactly solves the problem that CBC-like diffusion has low efficiency. Simulations have been carried out and analyzed in detail, proving the superior security and high efficiency of our image cryptosystem.

**INDEX TERMS** Image encryption, chaos, digit-level, block diffusion.

## I. INTRODUCTION

Nowadays, image security has attracted considerable attention as an increasing number of images are transmitted over the networks, shared in mobile phones and outsourced by the cloud storage. As encryption is the most popular technique for protecting privacy, many image encryption algorithms using different kinds of techniques [1]–[7] have been developed in the past decades. Among those techniques, chaotic systems which have many excellent intrinsic properties such as sensitivity to initial conditions, random-like dynamical behaviors, unpredictability and complex topological structures are appropriate for designing image cryptosystems.

In the last decade, a variety of chaotic systems [8], [9] have been employed to construct image cryptosystems, such as 2D logistic map [10], 2D standard map [11], 2D henon map [12], 3D LCA map [13], 3D cat map [14], 3D baker map [15]. Besides, some chaotic systems combining with other methods like one-time keys [16], DNA computing [17], [18], perceptron model [19], cellular automata [20], have also been used to enhance the security of image cryptosystems.

Most of chaos-based image encryption schemes are generally composed of two main stages: permutation and diffusion. These two stages repeat for multiple times for the sake of obtaining a good security level.

The permutation stage is responsible for decreasing the strong correlation between pixels adjacent to each other. The methods of permutation can be divided into two categories: pixel-level and bit-level, according to the smallest processing element.

In the first category, a pixel is treated as the smallest scrambling element. For example, Wang *et al.* [21] proposed a new block image encryption algorithm. It uses a random growth technique to scramble the pixels of the sub-images with the Arnold cat map. By this method, the periodic drawback of Arnold cat map can be eliminated. Hua *et al.* [22] proposed a high-speed image scrambling method. It can change row and column positions of pixels simultaneously, and thus can efficiently reduce the strong correlation between neighbor pixels. Wang *et al.* [23] introduced two complex chaotic systems which have larger chaotic ranges and more complex behaviors. These two chaotic systems are used to generate random chaotic sequences, and the pixels of the image are scrambled in the two-dimensional plane according to the sorted chaotic sequences. Parvin *et al.* [24] adopted a new strategy of scrambling. The image pixels are shifted by row

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

and then by column with the chaotic sequence. This strategy is easy to implement and very efficient.

Most pixel-level permutations shuffle the image by changing the pixel positions without modifying pixel values, so the histograms of the permuted image and the original image are identical. Such pixel-level permutations are vulnerable to histogram attacks and chosen/known-plaintext attacks if they have no diffusions or bad diffusions [25], [26].

In the second category, bit is considered as the basic operating element and the pixel matrix of a plain-image is usually transformed into a binary matrix. For instance, Tao *et al.* [27] proposed a selective image encryption scheme based on bit-level because each bit of the pixel contributes differently to total information of an image. It only encrypts the four higher bits of each pixel and leaves the lower four bits unchanged. Then, researchers [28]–[30] presented some improved schemes in which two distinct strategies are applied to permute the higher four bits and the lower four bits. Liu and Wang [9] pointed out some weaknesses in [30], such as requiring square original image to encrypt, and permuting the bits only within each bit plane. Liu et al. also proposed a bit-level permutation and high-dimension chaotic map to encrypt color image. However, both of these methods take much longer time to run than the pixel-level scrambling method. Zhang *et al.* [31] analyzed the intrinsic features of the bit distribution and employed an expand-and-shrink strategy in the permutation stage with the purpose of reducing the high correlation among higher bits and making the distributions in each bit plane smooth. Chen *et al.* [32] showed that the intrinsic image features in bit-level proposed by Zhang are not applicable to medical images. By employing a nonlinear inter-pixel computing and swapping approach, uniform bit distribution as well as certain image diffusion performance can be simultaneously achieved in the permutation stage. In [33], a new 3D bit matrix permutation is proposed, in which the Chen system is used to develop a random visiting mechanism to the bit level of the plain-image. By combining the Chen system with a 3D Cat map in the permutation stage, a new mapping rule is designed to map one random position to another random position. Unlike pixel-level permutation, the bit-level permutation can change the position and value of a pixel simultaneously, so the histogram of permuted image is different with that of the original image. However, since each pixel corresponds to eight bits, the time consuming of bit-level permutation is eight times as much as the pixel-level permutation if same permutation method is applied to two levels.

The diffusion stage is in charge of changing pixel values so as to obtain the avalanche effect and resist the differential attack. Most of the proposed pixel-level diffusion operations are based on the CBC-like mode that each pixel depends on all previous processed pixels. For example, a serial diffusion strategy is used in [22]. It starts from the pixel at the first row and first column of an image, and spreads one by one according to the direction of the column, then moves to the next column. This strategy achieves a great diffusion effect, but it is time-consuming due to the serial operation. Cao *et al.* [34] adopted a pixel diffusion method that two-dimension pixels matrix is transformed into one-dimensional sequences and the diffusion operation is performed on the one-dimensional sequences. This method is not only time-consuming but also needs a lot of space to store the sequences. Ye and Huang [35] proposed a method of block diffusion, which greatly reduced the time required for diffusion. However, the diffusion effect is poor without pre-operation.

Considering that pixel-level permutation cannot change the histogram of the image and bit-level permutation is comparatively time-consuming, this paper first proposes a digit-level permutation which takes digit as the smallest processing element. During the permutation, the pixel matrix of the plain image is decomposed into three matrices at the digit-level. Then, each matrix is scrambled for three rounds using Henon map with different control parameters. Finally, the three scrambled matrices are recombined into a new pixel matrix to obtain the permuted image. This method can not only change the histogram of the plain image after permutation, but also be more efficient than the bit-level permutation. Besides, in order to improve the efficiency of diffusion operation, a fast block diffusion strategy is proposed in this paper. The diffusion is serial in each block and parallel among different blocks. As a result, the diffusion process would be significantly accelerated if there is more than one processing unit. Theoretical and numerical analysis show that the proposed method has high security level and owns low time complexity.

The rest of this paper is organized as follows. In Section II, two-dimensional Henon map and 2D-LASM are introduced. In Section III, the proposed image encryption is described in detail. Simulation results and security analysis are presented in Section IV and V, respectively. Finally, Section VI concluded the paper.

## II. PRELIMINARIES

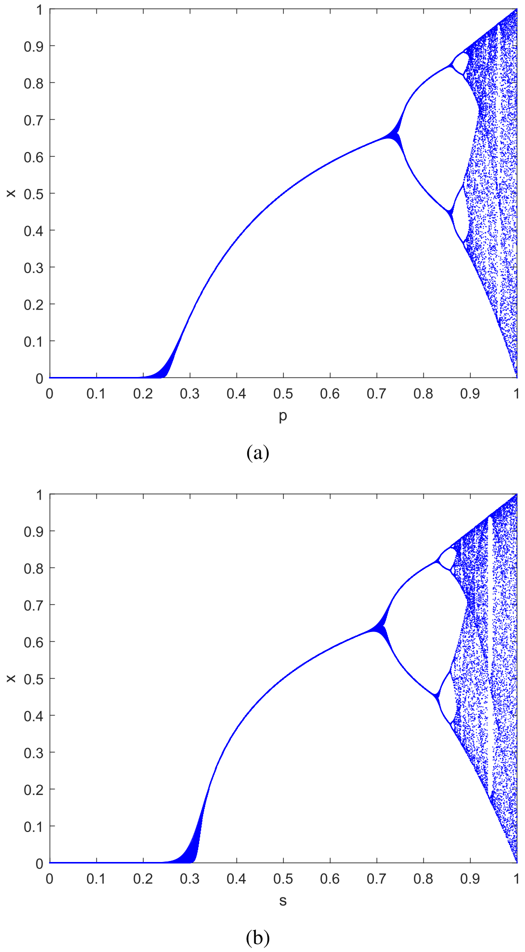
### A. TWO-DIMENSIONAL LOGISTIC-ADJUSTED-SINE MAP

The security of image encryption algorithms largely depends on the chaotic performance of the chaotic systems used. Logistic map and Sine map which are classical nonlinear discrete-time dynamical systems are often used in image encryption algorithms. They are defined by

$$x_{i+1} = 4px_i\,(1 - x_i) \tag{1}$$

$$x_{i+1} = s\sin(\pi x_i), \tag{2}$$

where parameters $p$ and $s$ are within the range of $[0, 1]$. Fig.1 shows the bifurcation diagrams of them. These two systems have chaotic behavior for parameter values $p \in [0.89, 1]$ and $s \in [0.87, 1]$. Although these two one-dimensional chaotic systems are easy to implement, their common drawbacks include relative simple chaotic trajectory, small key space, and easy predicable trajectory. At the same time, high-dimensional chaotic systems [4], [36] having complex chaotic behaviors are proposed, but these systems are relatively time-consuming. Hua and Zhou [3] proposed a



(a)

(b)

**FIGURE 1.** Bifurcations of chaotic map. (a) Logistic map; (b) Sine map.

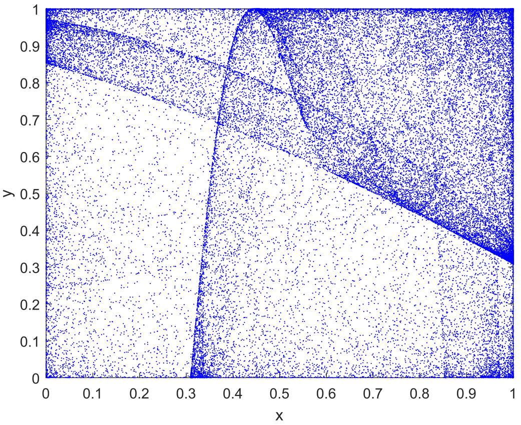


**FIGURE 2.** The output trajectory of 2D-LASM when $\mu = 0.9$.

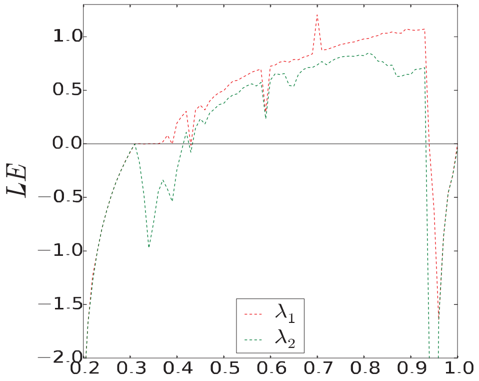


**FIGURE 3.** Lyapunov exponents of the 2D-LASM.

2D-Logistic-adjusted-Sine chaotic map (2D-LASM) that derives from Sine map and Logistic map. The formula of 2D-LASM is defined by

$$\begin{cases} x_{i+1} = \sin(\pi\mu(y_i+3)x_i(1-x_i)) \\ y_{i+1} = \sin(\pi\mu(x_{i+1}+3)y_i(1-y_i)), \end{cases} \tag{3}$$

where the parameter $\mu \in [0, 1]$. Fig.2 shows the output trajectory of the 2D-LASM when $\mu = 0.9$. We can see that the trajectory of 2D-LASM spreads over the whole two-dimensional plane and the chaotic effect is very great. Compared with Logistic map and Sine map, 2D-LASM has larger key space and more complex behavior. The Lyapunov exponent(LE) is a widely accepted indictor to evaluate the chaotic behavior of a dynamical system. LE measures the degree of divergence between two close trajectories of a dynamical system. A positive LE means that no matter how close the two trajectories are, with continuous iteration, their differences increase between the two trajectories until they are completely separated. Therefore, a dynamical system with a positive LE is chaotic. As a 2D discrete chaotic map, 2D-LASM has two LEs. Fig.3 presents how their LE values $\lambda_1$ and $\lambda_2$, change with respect to the corresponding control parameters. We can observe that 2D-LASM has chaotic behavior when $\mu \in [0.37, 0.38] \cup [0.4, 0.42] \cup [0.44, 0.93] \cup \{1\}$.

### B. TWO-DIMENSIONAL HENON MAP

In 1976, Hénon [37] first proposed the Henon map, and Henon map was extensively studied. The Henon map defined on a two-dimensional plane is a nonlinear discrete-time dynamical system. The formula is as follows:

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n. \end{cases} \tag{4}$$

Here, $x, y$ are iteration values while $n = 0, 1, 2 \ldots$ represents the number of map iterations. $a$,$b$ are two control parameters that the map depend on. For classical Henon map defined by Eq.(4), it can be seen that the iteration time is discrete while the iteration value is continuous. In [38], the Henon map is discretized so that its iterated values are generated over an integer domain. After discretization, the Henon map becomes

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \mod N \\ y_{n+1} = x_n + d \mod N, \end{cases} \tag{5}$$

where $x, y \in \{0, 1, 2, \cdots, N-1\}$, $a, d \in \{1, 2, \ldots, 2^{128}\}$ are two control parameters and $N$ represents the size of

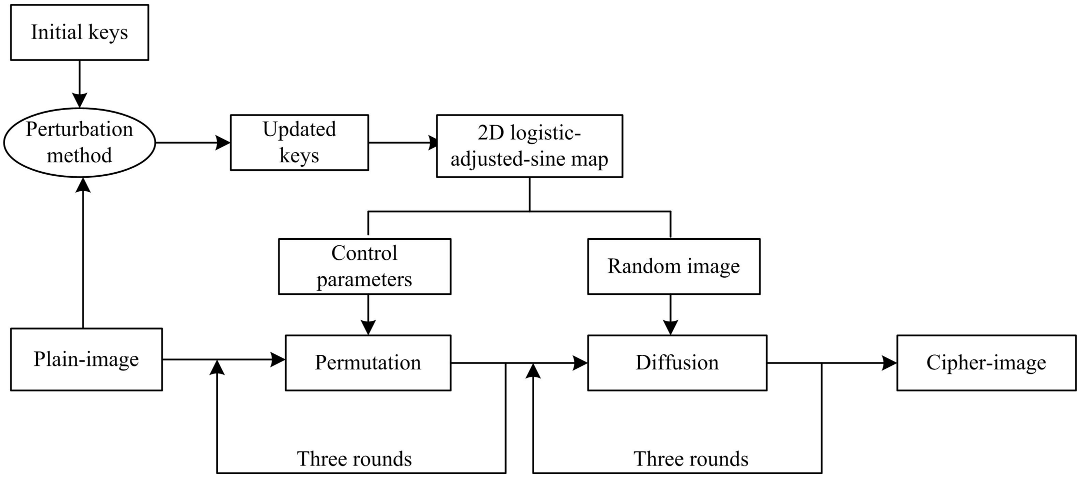


**FIGURE 4. The flow chart of the proposed image encryption algorithm.**

the image matrix. It should be noted that the image to be processed must be a square when Henon map is applied to image scrambling. Since the discrete Henon map is a one-to-one map for integers, it is efficient and easy to apply Henon map for image scrambling.

## III. DESCRIPTION OF THE PROPOSED CRYPTOSYSTEM

The proposed image cryptosystem employs the classic ‘permutation-diffusion’ structure. In the permutation stage, a new permutation method based on digit-level is proposed. Such digit-level permutation can eliminate high correlation among adjacent pixels as well as change the distribution of pixels of the plain-image. In the diffusion stage, a fast diffusion method was presented to solve the inefficiency of CBC-like diffusion mode. The flow chart of the proposed image cryptosystem is shown in Fig.4.

### A. GENERATION OF THE RANDOM-LIKE IMAGE AND CONTROL PARAMETERS

In order to resist the chosen-plaintext attack, the sub-key stream (control parameters and random-like image) used in the permutation and diffusion are generated by both 2D-LASM and plain-image. The detailed generation of the random-like image and the control parameters of the Henon map are as follows:

Input: The initial values $x_0$, $y_0$ of 2D Logistic-adjusted-Sine map, coefficient $\mu$ and plain-image $P = \{p(i,j)|1 \leq i \leq N, 1 \leq j \leq N\}$

Output: Random-like image $RI = \{ri(i,j)|1 \leq i \leq N, 1 \leq j \leq N\}$, the control parameters $a_0$, $a_1$, $a_2$, $b_0$, $b_1$, $b_2$ of the Henon map

Step 1: Calculate two plain-image features $val1$, $val2$ as follows

$$\begin{cases} val1 = \sum_{i,j} p(i,j) \mod 256 \\ val2 = \sum_{i,j} (p(i,j)+i)(p(i,j)-j) \mod 256 \end{cases} \tag{6}$$

Step 2: Calculate the perturbation coefficients $\alpha_1$, $\alpha_2$ by

$$\begin{cases} \alpha_1 = \dfrac{(val1+1)(val2+1)}{257^2} \\ \alpha_2 = \dfrac{(val1+2)(val2+2)}{258^2} \end{cases} \tag{7}$$

Step 3: Update the initial values $x_0$, $y_0$ of 2D Logistic-adjusted-Sine map by

$$\begin{cases} x_0 = x_0 \pm (\alpha_1 \times 10^5 - \ulcorner \alpha_1 \times 10^5 \urcorner) \times 10^{-5} \\ y_0 = y_0 \pm (\alpha_2 \times 10^5 - \ulcorner \alpha_2 \times 10^5 \urcorner) \times 10^{-5} \end{cases} \tag{8}$$

In the above perturbation method, the symbol “⌜⌝” represents the floor function. The symbol “±” means “+” or “−”. Here, we choose the operation “+”, if the updated initial condition is still in area; otherwise, the operation “−” is used instead.

Step 4: Iterate the 2D Logistic-adjusted-Sine map $N \times N+3$ times using the $x_0$ and $y_0$ obtained in step 3, and generate the chaotic sequences $x_1, x_2, \ldots, x_{N\times N}, x_{N\times N+1}, x_{N\times N+2}, x_{N\times N+3}$ and $y_1, y_2, \ldots, y_{N\times N}, y_{N\times N+1}, y_{N\times N+2}, y_{N\times N+3}$.

Step 5: A random-like image is generated as follows

$$ri(i,j) = (x_{(i-1)\times N+j} + y_{(i-1)\times N+j}) \times 10^{14} \mod 256 \qquad 1 \leq i \leq N, \quad 1 \leq j \leq N \tag{9}$$

Step 6: Generate the control parameters $a_0$, $a_1$, $a_2$, $b_0$, $b_1$, $b_2$ of the Henon map using Eq.(10).

$$\begin{cases} a_k = x_{N\times N+k} \times 10^{14} \mod 256 \\ b_k = y_{N\times N+k} \times 10^{14} \mod 256, \ k = 1,2,3 \end{cases} \tag{10}$$

### B. IMAGE PERMUTATION

Permutation operation is an important part of image encryption. It shuffles the position of the pixels in the plain-image, which can reduce the correlation of adjacent pixel values. The scrambling method proposed in this paper takes the digit as the smallest processing element. First, the pixel matrix of plain-image is decomposed into three matrices. Then, these

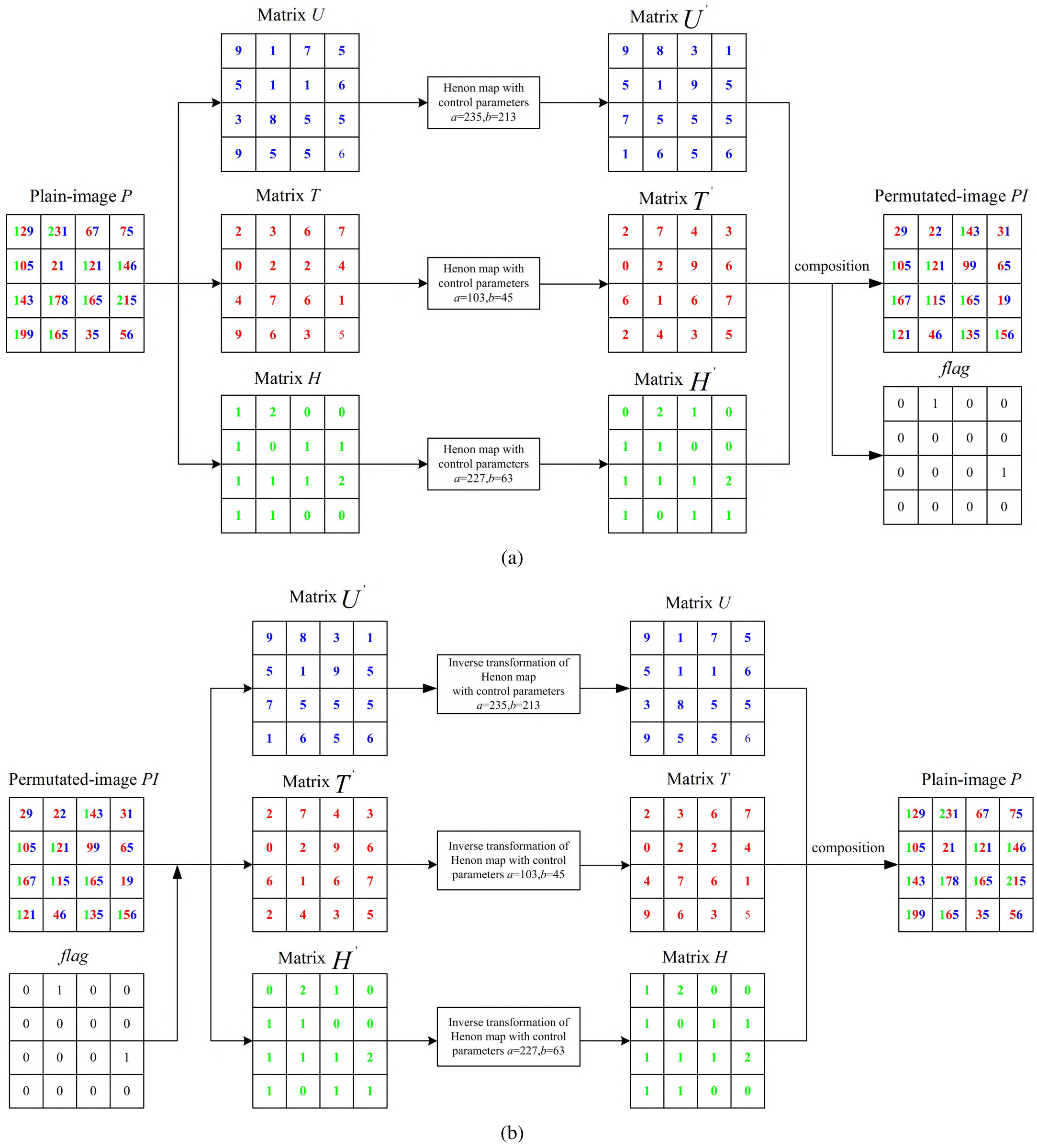

Plain-image P
Matrix U
Matrix T
Matrix H
Henon map with control parameters a=235,b=213
Henon map with control parameters a=103,b=45
Henon map with control parameters a=227,b=63
Matrix U'
Matrix T'
Matrix H'
composition
Permutated-image PI
flag
(a)
Permutated-image PI
flag
Matrix U'
Matrix T'
Matrix H'
Inverse transformation of Henon map with control parameters a=235,b=213
Inverse transformation of Henon map with control parameters a=103,b=45
Inverse transformation of Henon map with control parameters a=227,b=63
Matrix U
Matrix T
Matrix H
composition
Plain-image P
(b)

**FIGURE 5.** An example of permutation algorithm: (a) Permutation, (b) Reverse permutation.

three matrices are respectively scrambled by Henon map with different control parameters. Finally, these scrambled matrices are composed into a new matrix. Fig.5 shows an example of the proposed permutation.The detailed steps of the permutation are as follows:

Input: Plain-image $P = \{p(i,j)|1 \leq i \leq N, 1 \leq j \leq N\}$, control parameters $a_0, a_1, a_2, b_0, b_1, b_2$

Output: Permutated image $PI = \{pi(i,j)|1 \leq i \leq N, 1 \leq j \leq N\}$ and mark-matrix *flag*

Step 1 : The pixel matrix of plain-image $P$ is decomposed into three matrices $U, T, H$ according to Eq.(11).

$$\begin{cases} U(i,j) = \text{mod}(P(i,j), 10) \\ T(i,j) = fix(\text{mod}(P(i,j)/10, 10)) \\ H(i,j) = fix(P(i,j)/100) \end{cases} \tag{11}$$

Here, *fix* is a function to round the argument to zero.

Step 2: Scramble matrices $U, T, H$ with Henon map and obtain three new permuted matrices $U'$, $T'$ and $H'$. For each matrix, calculate the new position $(i', j')$ by Eq.(12), and then move the element $(i, j)$ to $(i', j')$. Here, the control parameters of Henon map $(a_0, b_0)$, $(a_1, b_1)$, $(a_2, b_2)$ are used for $U, T, H$ respectively.

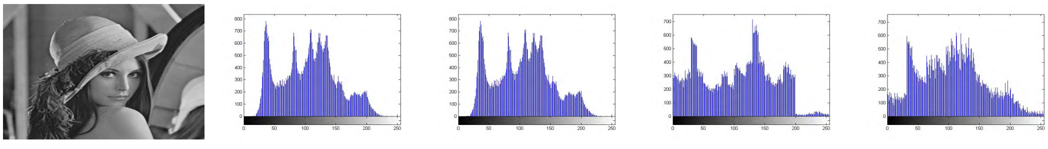
$$\begin{cases} i' = 1 - ai^2 + j \mod N \\ j' = i + b \mod N \end{cases} \tag{12}$$

Step 3: Repeat the scrambling operation in step 2 twice to generate new matrices $U'$, $T'$ and $H'$.

Step 4: The three new matrices $U'$, $T'$ and $H'$ obtained in step 3 are composed simultaneously to generate a new matrix $PI$ and mark-matrix *flag* according to Eq.(13) and

**TABLE 1.** Execution time comparison between proposed permutation method and other available permutation methods.

| Permutation | Rounds | Correlation of pixels | | | Time(ms) |
|---|---|---|---|---|---|
| | | Horizontal | Vertical | Diagonal | |
| Proposed(digit-level) | 3 | -0.0004 | 0.0007 | 0.0077 | 68 |
| Ref. [39](pixel-level) | 3 | 0.0202 | -0.0062 | -0.0106 | 24 |
| Ref. [9](bit-level) | 3 | -0.0035 | -0.0574 | 0.0578 | 327 |



(a) (b) (c) (d) (e)

**FIGURE 6.** The simulation results: (a) Plain image, (b) Histogram of the plain image, (c) Histogram of the pixel-level based on scrambled image, (d) Histogram of the digit-level based on scrambled image, (e)Histogram of the bit-level based on scrambled image.

Eq.(14); the mark-matrix *flag* is used to mark the position where the value of every position in the matrix *PI* exceeds 256.

$$pi(i,j) = U''(i,j) + 10 \times T''(i,j) + 100 \times H''(i,j) \quad (13)$$

$$flag(i,j) = \begin{cases} 1, & \text{if } pi(i,j) \geqslant 256 \\ 0, & \text{if } pi(i,j) < 256 \end{cases} \quad (14)$$

Step 5: Matrix *PI* is updated by:

$$pi(i,j) = pi(i,j) \mod 256, \quad \text{if } pi(i,j) \geqslant 256 \quad (15)$$

Here, we compare the pixel correlation and the execution time of the proposed method with other permutation methods. The tested image is gray Lena with 256 × 256 pixels. In Table 6, three permutation methods based on digit-level, pixel-level [39] and bit-level [9] are used for comparison. It shows that the ability of our digit-level permutation for removing correlation is better than that of Mollaeefar's [39] and Liu's [9]. For speed test, the pixel-level permutation [39] is faster than digit-level permutation and bit-level permutation [9]. This is because pixel-level permutation needs to scramble $256 \times 256$ pixels, and digit-level permutation needs to scramble 256 × 256 × 3 digits, and bit-level permutation has to scramble 256 × 256 × 8 bits. Fig.6 shows the histograms of the original image and the permutated images for the three algorithms. It can be seen that the proposed digit-level permutation and bit-level permutation in [9] can both change the distribution of pixel values while pixel-level permutation [39] cannot do this. The scrambling capability of bit-level is almost the same as that of the proposed digit-level. However, the proposed algorithm can obtain faster speed and lower pixel correlation.

### C. IMAGE DIFFUSION

Diffusion operation is the most important stage in image encryption algorithms. It not only changes the distribution of pixel values but also has an avalanche effect that if a pixel value changes, it will lead to the entire cipher-image changed. The diffusion method proposed in this paper combines block diffusion with CBC-like mode, which is efficient and achieves excellent performance. Fig.7 describes the diffusion operation, a detailed description of the proliferation steps are as follows:

Input: Permutated image $PI = \{pi(i,j)|1 \leq i \leq N, 1 \leq j \leq N\}$, random-like image $RI = \{ri(i,j)|1 \leq i \leq N, 1 \leq j \leq N\}$

Output: Cipher-image $C = \{c(i,j)|1 \leq i \leq N, 1 \leq j \leq N\}$

Step 1: Matrices *PI* and *RI* are divided into *N* row vectors $PI = \{PV_1, PV_2, \ldots, PV_N\}$, $RI = \{RV_1, RV_2, \ldots, RV_N\}$.

Step 2: Compute and update each row vector $PV_i$ according to Eq.(16). The updated steps start from the second row vector $PV_2$ and end at the first row vector $PV_1$.

$$PV_i' = \begin{cases} PV_i + PV_{i-1} + RV_i \mod 256, & \text{if } i = 2, 3, \ldots, N \\ PV_i + PV_N + RV_i \mod 256, & \text{if } i = 1 \end{cases} \quad (16)$$

Step 3: Update the first row vector $PV_1'$ twice according to the Eq.(17), and obtain $PV_1'' = \{e_1', e_2', \ldots e_N'\}$. Here, $PV_1' = \{e_1, e_2, \ldots, e_N\}$ and $RV_1 = \{re_1, re_2, \ldots, re_N\}$.

$$e_i' = \begin{cases} e_i + e_{i-1} + re_i \mod 256, & \text{if } i \geqslant 2 \\ e_i + e_N + re_i \mod 256, & \text{if } i = 1 \end{cases} \quad (17)$$

Step 4: Compute the vector set $PI' = \{PV_1, PV_2, \ldots, PV_N\}$, according to the Eq.(18).

$$PV_k = \begin{cases} PV_k', & k = 2, 3, \ldots, N \\ PV_k'', & k = 1 \end{cases} \quad (18)$$

Step 5: Take $PI'$ as the new input, and then repeat step 1 through 4 again to get $PI'' = \{PV_1, PV_2, \ldots, PV_N\}$, finally, the encrypted image $C = \{CV_1, CV_2, \ldots, CV_N\}$ is formed according to the Eq.(19).

$$CV_i = PV_i, i = 1, 2, 3, \ldots, N \quad (19)$$

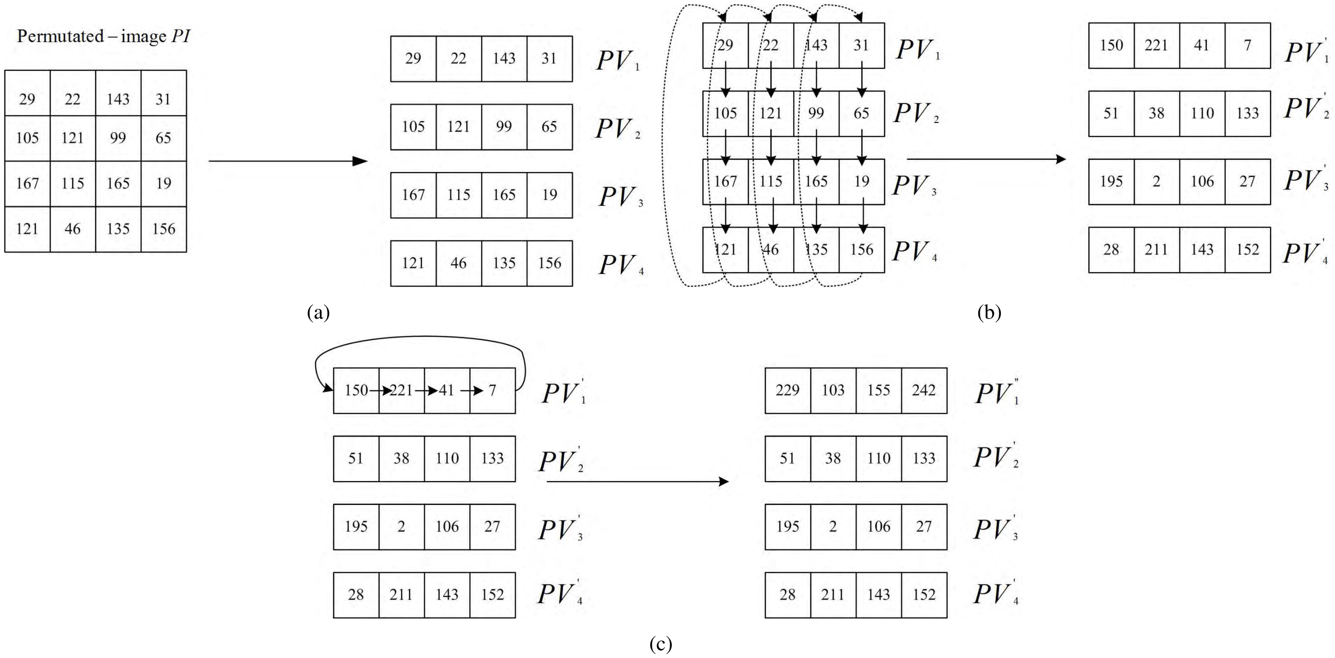


FIGURE 7. An example of the diffusion: (a) Step 1 of the diffusion, (b) Step 2 of the diffusion (c) Step 3 of the diffusion.

### D. DESIGN OF THE DECRYPTION SCHEME

The decryption phase is the inverse process, and the steps are as follows:

Input: The encrypted image, the initial values $x_0$, $y_0$, two plain-image features $val1$, $val2$, the mark-matrix $flag$

Output: The decrypted image

Step 1: Use initial values $x_0$, $y_0$ and plain image features $val1$, $val2$ to calculate the updated values $x_0$, $y_0$ by Eq.(7) and Eq.(8).

Step 2: Iterate the 2D-LASM with the updated $x_0$, $y_0$ to generate two chaotic sequences by Eq.(3). The random-like image and control parameters of Henon map are achieved from the chaotic sequences by Eq.(9) and Eq.(10).

Step 3: Calculate twice using Eq.(20), and get the first row vector, and then use the first row vector and formula (21) to calculate the other row vectors of the vector set.

$$e_i = \begin{cases} e'_i - e_{i-1} - re_i \mod 256, & \text{if } i \geqslant 2 \\ e'_i - e_N - re_i \mod 256, & \text{if } i = 1 \end{cases} \tag{20}$$

$$PV_i = \begin{cases} PV'_i - PV_{i-1} - RV_i \mod 256, & \text{if } i = 2, 3, \ldots, N \\ PV'_i - PV_N - RV_i \mod 256, & \text{if } i = 1 \end{cases} \tag{21}$$

Step 4: Repeat step 3 again to generate permutated image.

Step 5: Recover the pixels of the permutated image whose values exceed 256 by using the mark-matrix $flag$.

Step 6: Decompose the pixel matrix of permutated image into three matrices by (11), and then scramble these three matrices by

$$\begin{cases} i = j' - b \mod N \\ j = i' - 1 + ai^2 \mod N \end{cases} \tag{22}$$

Step 7: Repeat step 6 twice to obtain the plain-image.

## IV. SIMULATION RESULTS

The results of the extensive computer simulations have been done with the Matlab 2014 programming environment. This paper selected four images($256 \times 256$) with different distribution of pixel values for encryption from the image database CVG-UGR. Fig.8(a) and Fig.8(b) show the original images and their histograms. Fig.8(c) and Fig.8(d) depict the permutated images and their histograms. We can see that the histograms of the permutated images are different from that of the original images, which is able to prevent the opponent from doing any statistical analyses. Fig.8(e) and Fig.8(f) show the cipher-images obtained by the proposed scheme and their histogram. It is clear that the encrypted images are noise-like and we can not find any visual feature of original images in the encrypted images. Additionally, the histograms of the cipher-images are all quite uniform which can cover the original images effectively.

## V. PERFORMANCE AND SECURITY ANALYSIS

Several experiments and different kinds of security analyses are performed to evaluate the robustness of the proposed algorithm in these sections.

### A. SECRET KEY SPACE ANALYSIS

For the image encryption algorithm, the set of all possible security keys form the key space of the algorithm. To resist the brute-force attack, the key space should be at least $2^{100} \approx 10^{30}$ [40], [41]. In our proposed algorithm, $x_0$, $y_0$ and $\mu$ of the 2D-LASM are the security keys. Considering the computational precision of each key can reach $10^{-14}$,

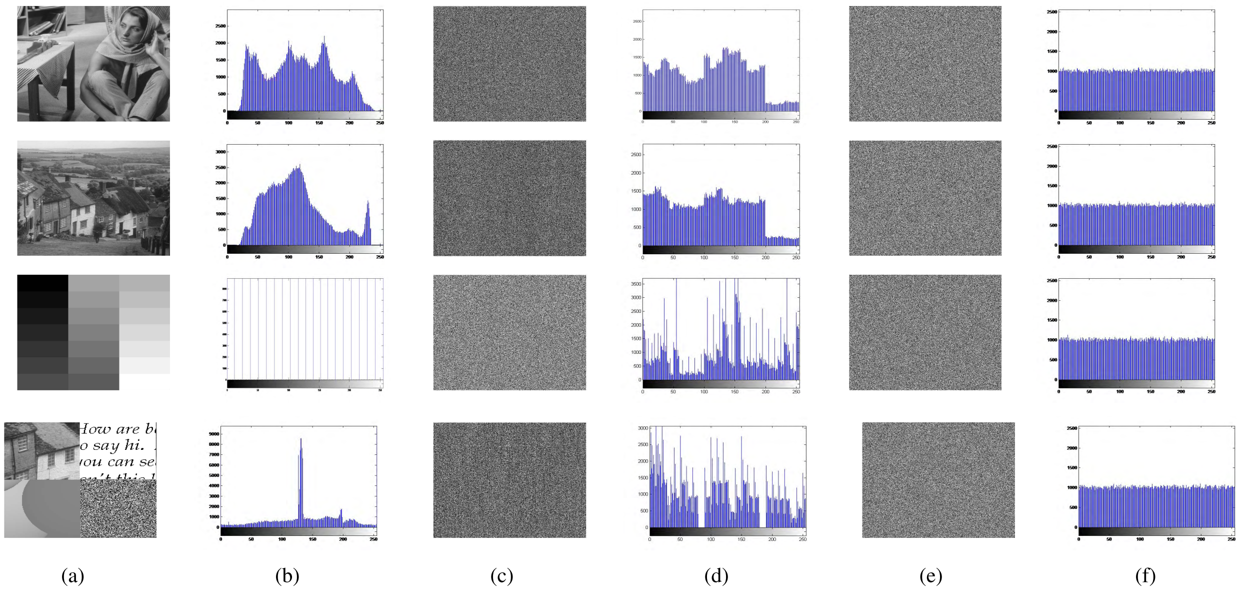


**FIGURE 8. The simulation results: (a) Plain images, (b) Histograms of the plain images, (c) Permutated images, (d) Histograms of the permutated images, (e) Cipher images, (f) Histograms of the cipher images.**

the entire key space size can reach $10^{42}$ which prohibits exhaustive search of the key space.

### B. RANDOMNESS TEST

The randomness of the output sequence is one of them criteria for evaluating the security of cryptographic algorithms. Although various kind of randomness tests have been proposed so far, there is no known sequences generated by an algorithm which pass all the randomness tests. One of the most popular tests of randomness SP800-22 [42] is published by the national institute of research Standard sand technology (NIST). The SP800-22 provided a statistical test suite (STS) consisting of 15 tests. These tests focus on a variety of different types of non-randomness that could exist in one sequence. As recommended by NIST, a significance level $\alpha = 0.01$ was used for test. For each sub-test, the $P-value$ is computed for all binary sequences and compared with significance level $\alpha$. A binary sequence is regarded as passing a sub-test if $P-value \geq \alpha$ and failing otherwise.

NIST recommended two approaches to interpret test results: First, to calculate the pass rate of sequences passing a sub-test. If the pass rate falls outside the scope of acceptable proportion defined by Eq.(23), then it reveals that the test sequence is not random enough. Second, to caculate the distribution of $P-Values$ and examine whether or not the $P-values$ are uniform distribution in the range $[0, 1]$. By dividing range $[0, 1]$ into ten equal sub-intervals, $F_i$ denotes the frequency of the $P-values$ falling inside of $i$-th sub-interval. Let $m$ be the sample size and the $P-value_T$ is computed by Eq.(24) and Eq.(25). A sequence could be considered to be uniform distribution if $P-value_T \geq 0.0001$.

$$\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}, \quad where\ \hat{p} = 1-\alpha. \tag{23}$$

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - m/10)^2}{m/10}. \tag{24}$$

$$P-value_T = igamc\langle \frac{9}{2}, \frac{\chi^2}{2} \rangle \tag{25}$$

In our experiments, we iterated 2, 500, 000 times of the 2D-LASM chaotic system, generated two chaotic sequences, and converted the generated chaotic sequences into binary numbers to generate binary sequences $X$ and $Y$, respectively. The sequences $X$ and $Y$ are divided into 100 sets of binary sequences of 1, 000, 000 bits in length, and the 100 sets of binary sequences are subjected to NIST randomness tests. Table 2 provides NIST statistical results to test our proposed algorithm. The results show that the proposed algorithm can successfully pass all 15 tests. Therefore, we can have high randomness in our encrypted image scheme.

### C. INFORMATION ENTROPY ANALYSIS

Information entropy is an important analytical mmethod to measure the randomness of a message. Shannon proposed this method and its formula is defined by

$$IE(m) = \sum_{i=1}^{2^l-1} p(m_i) \log \frac{1}{p(m_i)} \tag{26}$$

Here, $l$ refers to the length of a pixel value in bit, $m$ is the message obtained and $p(m_i)$ represents the probability of $m_i$ occurrence; This paper selected two sizes 256 × 256 and 512 × 512 of four images for testing, the final test results shown in the Table 3.

For an ideal algorithm, it should have such a property that the ciphertext image is sufficiently random and the entropy of information is close to the theoretical 8. Table3 shows that

**TABLE 2.** Results of the NIST statistical tests for 2D-LASM.

| Statistical Test | $X$ sequence | | $Y$ sequence | |
|---|---|---|---|---|
| | $P-value_T$ | Proportion | $P-value_T$ | Proportion |
| Frequency | 0.685587 | 0.97 | 0.554420 | 1.00 |
| Block frequency($m = 20000$) | 0.678686 | 1.00 | 0.108791 | 1.00 |
| Runs(Forward) | 0.304126 | 0.99 | 0.145326 | 1.00 |
| longest runs of ones | 0.816537 | 1.00 | 0.236801 | 0.99 |
| Rank | 0.153763 | 0.99 | 0.867692 | 1.00 |
| Spectral DFT | 0.275709 | 0.97 | 0.494292 | 0.99 |
| Non-overlapping Templates($m = 9$) | 0.779188 | 1.00 | 0.455937 | 0.97 |
| Overlapping templates($m = 9$) | 0.924076 | 1.00 | 0.014550 | 0.99 |
| Maurers universal | 0.946308 | 0.96 | 0.366918 | 0.97 |
| Linear complexity($m = 500$) | 0.779188 | 1.00 | 0.474986 | 0.99 |
| Serial | 0.574903 | 1.00 | 0.224861 | 0.97 |
| Approximate entropy(m=10) | 0.983453 | 0.97 | 0.289667 | 0.99 |
| Cumulativesums(Forward) | 0.289667 | 0.97 | 0.262249 | 1.00 |
| Random excursions($x = -1$) | 0.548605 | 0.99 | 0.304126 | 1.00 |
| Random excursionsvariant($x = -1$) | 0.607399 | 1.00 | 0.554420 | 0.99 |

**TABLE 3.** Entropy test for different sizes of cipher images.

| | Name | Global entropy | Actual block entropy | Theoretical block entropy | |
|---|---|---|---|---|---|
| | | | | $\alpha = 0.01$ | $\alpha = 0.05$ |
| | | | | 7.16276745 | 7.16634107 |
| 256 × 256 | barche | 7.9971041 | 7.17835661 | Pass | Pass |
| | hedgebw | 7.99683937 | 7.17358089 | Pass | Pass |
| | leopard | 7.99719551 | 7.17983531 | Pass | Pass |
| | clock | 7.99654319 | 7.17593964 | Pass | Pass |
| mean | | 7.996921 | 7.176928 | | |
| 512 × 512 | 1 | 7.99926953 | 7.17770757 | Pass | Pass |
| | 2 | 7.99930523 | 7.1803292 | Pass | Pass |
| | 7 | 7.9992605 | 7.17289517 | Pass | Pass |
| | 49 | 7.9993729 | 7.1766752 | Pass | Pass |
| mean | | 7.999302 | 7.176902 | | |

the proposed algorithm is extremely close to the theoretical values.

## D. HISTOGRAM ANALYSIS

Histogram of the image shows the pixel values of the distribution information. The ideal encrypted image should have a uniform, disparate histogram to prevent the enemy from extracting any meaningful information from the wave histogram of the image. For the number analysis of each key, we use the histogram variance [43] to evaluate the consistency of the encrypted image. The lower the variance value, the higher the uniformity of the encrypted image. We also calculated the two variances of the encrypted images, which are encrypted by different keys on the same plain text image. The closer the two variance values are, the more consistent the encrypted image is when the key changes. The histogram variance is as follows:

$$var(Z) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(z_i - z_j)^2 \qquad (27)$$

where $Z$ is the vector of the histogram values and $Z = \{z_1, z_2, \ldots, z_{256}\}$, $z_i$ and $z_j$ are the numbers of pixels which gray values are equal to $i$ and $j$ respectively. In simulating experiments, we calculate two variances of histograms of two ciphered images by Eq.(27) from the same plaintext image with different secret keys. Only one parameter of secret keys is changed in such different secret keys. The plaintext images of hedgebw clock and leopard are tested for three round encryption in experiments. Table 4 lists the variances of histograms of ciphered hedgebw clock and leopard images. In Table 4, the variances in the first column are obtained by the secret key $key1$, the variances in next columns are obtained by only changing one parameter of $x$, $y$ and $\mu$ respectively compared with the secret key $key1$.

**TABLE 4.** Variances of histograms compared among all secret keys in the proposed algorithm.

| Ciphered image | $key1$ | $x$ | $y$ | $\mu$ |
|---|---|---|---|---|
| hedgebw | 52208.96 | 52457.3593 | 52457.3593 | 52767.8828 |
| clock | 74298.66 | 74009.789 | 73737.0781 | 74114.6718 |
| leopard | 85362.67 | 85394.3671 | 85158.625 | 85601.4609 |
| Average | 70623.43 | 70620.50513 | 70451.0208 | 70828.00517 |

## E. ATTACK ANALYSIS

According to the kerckhoff principle [44], it is generally assumed that attackers can obtain the design and work information of the password system studied when conducting password analysis on the password system, namely: For any researcher, he or she can know everything about the encryption system except the encryption and decryption keys.

This standard is the basic standard of any encryption system in the current secure communication network.There are four classical types of attacks:

(1) Ciphertext only: the attacker possesses a set of ciphertexts. The goal is to recover as much plaintexts as possible or even to guess the secret key.

(2) Known plaintext: the attacker possesses a set of the ciphertext and its corresponding plaintext. The goal is to recover the secret key or to decrypt any further messages.

(3) Chosen plaintext: the attacker has obtained temporary access to the encryption machinery. Hence, he can choose arbitrary plaintext to be encrypted and then receives the corresponding ciphertext. He tries to analyze and discover more key-related information.

(4) Chosen ciphertext: the arracker can analyze any chosen ciphertexts together with their corresponding plaintexts. The goal is to acruire the secret key or to get as many information about the encryption system as possible.

Among these attacks, Chosen plaintext attack is recognized as the most powerful attack. If a cryptosystem can resist the chosen plaintext attack, it also can resist other three kinds of attacks. In order to resist the chosen plaintext attack, we first calculate two plain-image features $val1$, $val2$ and then update the initial values of 2D LASM with these image features. As a result, the key stream (control parameters and random-like image) used in our permutation and diffusion strongly depends on the secret key and the plain-image. Since different plain-images will generate different key streams, the attacker cannot obtain any useful information about the secret key by a number of possible pairs of plain image and cipher image. Therefore, the proposed cryptosystem can resist the chosen plaintext attack.

### F. CORRELATION ANALYSIS

Generally, correlation between adjacent pixels in three directions is always high for most natural images. Therefore, the correlation of the adjacent pixels for encrypted cipher-image is one of an important criteria to measure the performance of the algorithm. Here, Eq.(28) and Eq.(29) are the formulas to calculate the correlation between two adjacent pixels [45].

$$\gamma = \frac{\sum_{i=1}^{M}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{M}(x_i - E(x))^2 \times \sum_{i=1}^{M}(y_i - E(y))^2}} \tag{28}$$

$$E(x) = \frac{1}{M}\sum_{i=1}^{M} x_i, \tag{29}$$

Where, $x$ and $y$ are the gray values of adjacent pixels, $M$ refers to the total number of samples. In this test, we select five gray-scale images from the image database CVG-UGR, and randomly choose 20000 pairs of adjacent points in three directions for calculation, respectively. At the same time, in order to reflect the superiority of the proposed algorithm, we choose algorithm [46] and algorithm [47] to test and compare. Table5 gives the final results.

From Table5, the correlation coefficients of the five plain-images are very high, while the correlation coefficients of the encrypted images are almost close to zero. It shows that the encryption algorithm proposed in this paper can greatly reduce the correlation between adjacent pixels. In addition, through the data comparison, we can find that the proposed encryption algorithm is better than the other two algorithms.

### G. DIFFERENTIAL ANALYSIS

Differential attack aims to encrypt the plain-image with slight modification, and then find out the differences between two encrypted images. Comparing with the differences, the attacker can discover the relationship between plain-image and cipher-image. Two well-known parameter values

**TABLE 5.** Correlation coefficient comparison between nine different images and their corresponding encrypted images.

| Test Image | Direction | Plain image | Cipher image | | |
|---|---|---|---|---|---|
| | | | Proposed | Ref. [47] | Ref. [46] |
| Lena | Horizontal | 0.946421 | -0.000842 | -0.000187 | -0.002045 |
| | Vertical | 0.972770 | -0.002503 | 0.001835 | -0.002845 |
| | Diagonal | 0.922830 | 0.001067 | -0.000310 | -0.001220 |
| Barche | Horizontal | 0.940052 | 0.001771 | 0.002178 | -0.002189 |
| | Vertical | 0.942132 | 0.002445 | -0.003358 | 0.001205 |
| | Diagonal | 0.896352 | 0.001449 | 0.004433 | -0.007354 |
| Clock | Horizontal | 0.957544 | 0.003344 | 0.003748 | 0.000891 |
| | Vertical | 0.974313 | -0.002060 | 0.003066 | 0.003765 |
| | Diagonal | 0.940198 | -0.001137 | -0.003784 | 0.001458 |
| Hedgebw | Horizontal | 0.952767 | -0.000878 | -0.009464 | 0.006559 |
| | Vertical | 0.948055 | -0.006793 | -0.005243 | -0.007743 |
| | Diagonal | 0.912408 | 0.001309 | -0.007416 | -0.007396 |
| Leopard | Horizontal | 0.904922 | 0.001581 | -0.001394 | 0.002242 |
| | Vertical | 0.930216 | -0.001017 | 0.003244 | 0.002828 |
| | Diagonal | 0.873007 | -0.000172 | -0.003250 | 0.000232 |
| Average (absolute values) | Horizontal | 0.940341 | 0.0016832 | 0.003394 | 0.0027852 |
| | Vertical | 0.953497 | 0.0029636 | 0.0033492 | 0.0036772 |
| | Diagonal | 0.908959 | 0.0010268 | 0.0038386 | 0.003532 |

**TABLE 6.** UACI and NPCR results for different image encryption algorithms.

| Image | | Image 256 × 256 | | | Image512 × 512 | | | Image1024 × 1024 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | min | max | mean | min | max | mean | min | max | mean |
| Proposed | NPCR | 99.5438 | 99.6658 | 99.6126 | 99.5846 | 99.6368 | 99.6106 | 99.5885 | 99.6294 | 99.6088 |
| Ref. [35] | NPCR | 98.4875 | 98.8902 | 98.829 | 99.1937 | 99.2537 | 99.2199 | 99.258 | 99.3652 | 99.3504 |
| Ref. [47] | NPCR | 49.7772 | 99.6414 | 94.1226 | 49.8024 | 99.6449 | 98.1192 | 49.7921 | 99.6301 | 92.6337 |
| Proposed | UACI | 33.2253 | 33.7013 | 33.4466 | 33.3125 | 33.594 | 33.463 | 33.3836 | 33.5661 | 33.4669 |
| Ref. [35] | UACI | 32.9306 | 33.42382 | 33.1919 | 33.19 | 33.4071 | 33.3177 | 33.3153 | 33.4809 | 33.3778 |
| Ref. [47] | UACI | 16.6759 | 33.66 | 32.3196 | 16.7439 | 33.5757 | 32.9797 | 16.7155 | 33.5134 | 31.1178 |

**TABLE 7.** NPCR performance of different rounds.

| Round | Proposed | Ref. [48](sp and sp1) | Ref. [48](sp and sp2) | Ref. [48](sp and sp3) |
|---|---|---|---|---|
| 1 | 0.9960606384 | 0.995754242 | 0.995758213 | 0.995769500 |
| 2 | 0.9961079407 | 0.995769500 | 0.995766432 | 0.995773315 |
| 3 | 0.9961264854 | 0.996654437 | 0.996654353 | 0.996446013 |

**TABLE 8.** UACI performance of different rounds.

| Round | Proposed | Ref. [48](sp and sp1) | Ref. [48](sp and sp2) | Ref. [48](sp and sp3) |
|---|---|---|---|---|
| 1 | 0.3346300551 | 0.167879277 | 0.167879365 | 0.168504374 |
| 2 | 0.3346130066 | 0.273548436 | 0.273558654 | 0.284346542 |
| 3 | 0.3344662753 | 0.334285344 | 0.334286401 | 0.334305632 |

called Number of Pixels Change Rate(NPCR) and Unified Average Changing Intensity (UACI) [46] are used to measure the ability of image encryption algorithms to resist differential attacks. Eq.(30) Eq.(31) and Eq.(32) are the calculated formulas.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%, \tag{30}$$

$$UACI = \frac{1}{M \times N}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%, \tag{31}$$

$$D(i,j) = \begin{cases} 0 & \text{if } C_1(i,j) = C_2(i,j); \\ 1 & \text{if } C_1(i,j) \neq C_2(i,j), \end{cases} \tag{32}$$

where $M$ and $N$ denote the size of the image, $C_1$ and $C_2$ are two ciphertext images, which differ only in one pixel position. $C_1(i,j)$ and $C_2(i,j)$ are gray-values of the pixels at position $(i,j)$ of two ciphertext images. In this paper, Images with three different size $128 \times 128$, $256 \times 256$, $512 \times 256$ were selected for testing with the same secret key. Moreover, the final result values of each image generated after 100 times tests which extract the maximum values, minimum values and the average values. In addition, algorithm [46] and algorithm [47] are selected for performance comparison, the final results are shown in Table6.

Table6 shows that the mean values of NPCR and UACI produced by our proposed algorithm is extremely close to the expected values $NPCR_{Expected} = 99.61\%$ and $UACI_{Expected} = 33.46\%$. Besides, the minimum and maximum NPCR and UACI are more close to the mean value. These indicate that the proposed algorithm can effectively resist the difference attack.

In addition, we selected the Zhang's algorithm presented in [48] as a comparison test for the different rounds of encryption. Zhang and Wang [48] used four plain images: the original plain image Lena $sp$, the image $sp1$ only changing one bit in the pixel $sp[511, 511]$, the image $sp2$ only changing one bit in the pixel $sp[0, 0]$, and the image $sp3$ only changing one bit in the pixel $sp[256, 256]$. The four images are then encrypted with the same key.

Table7 and table8 show that the proposed method requires only one encryption round to obtain the expected values of NPCR and UACI. Zhang's algorithm [48] also needs only one round of encryption to acquire the ideal value of NPCR, while it needs at least three rounds of encryption to obtain the ideal value of UACI. Therefore, the method proposed in this paper is slightly more efficient.

### H. TIME COMPLEXITY ANALYSIS

The speed of image encryption algorithms is also an important indicator of algorithmic performance. Especially in the era of big data, the encryption speed of the encryption algorithm is especially important for the demand of image encryption with large amount of data. This paper selects three sizes of images for time complexity analysis. The experimental platform used in this experiment is configured as: i7CPU (2.60GHz), memory 8G, operating system win7 (64bit); This paper also selected two kinds of image encryption algorithms [47] and [46] for comparison, the final test result are given in 9. It shows that the proposed algorithm is faster than other two algorithms. Especially in diffusion step, the larger the size of the image, the faster the diffusion.

**TABLE 9.** Encryption speed(seconds) for different size of images.

| Image size | Proposed | Ref. [3] | Ref. [47] | Ref. [46] |
|---|---|---|---|---|
| Image 128 × 128 | 0.038122 | 0.081401 | 0.105083 | 0.348629 |
| Image 256 × 256 | 0.091381 | 0.124262 | 0.245695 | 0.323390 |
| Image 512 × 512 | 0.212328 | 0.382367 | 0.902855 | 0.900496 |
| Image 1024 × 1024 | 0.745312 | 1.424795 | 3.433845 | 3.024380 |

## VI. CONCLUSION

This paper presents a new image encryption scheme based on digit-level permutation and block diffusion. The digit-level permutation can change the histogram of the image, so it is difficult for attackers to do any statistical analyses because the histogram is changed greatly after permutation. The block diffusion method is extremely efficient and can obtain the ideal values of NPCR and UACI. Additionally, the proposed image encryption algorithm also adopts the image feature to update the initial values of the 2D-LASM. As a result, the keystream generation is dependent on the plain-image which can effectively resist the chosen-plaintext attack. The experimental results prove that the proposed scheme has the ability to encrypt digital images into random-like cipher-images and resist common attacks effectively, which is very suitable for image encryption.

## REFERENCES

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

[2] H. Zhu, C. Zhao, X. Zhang, and L. Yang, "An image encryption scheme using generalized Arnold map and affine cipher," *Optik*, vol. 125, no. 22, pp. 6672–6677, 2014.

[3] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.

[4] Y. Li, C. Wang, and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Opt. Lasers Eng.*, vol. 90, pp. 238–246, Mar. 2017.

[5] A. Belazi, A. A. A. El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Process.*, vol. 128, pp. 155–170, Nov. 2016.

[6] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using dna sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, Jan. 2017.

[7] J. Kalpana and P. Murali, "An improved color image encryption based on multiple DNA sequence operations with DNA synthetic image and chaos," *Optik*, vol. 126, no. 24, pp. 5703–5709, 2015.

[8] X. Huang and G. Ye, "An efficient self-adaptive model for chaotic image encryption algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 12, pp. 4094–4104, 2014.

[9] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Opt. Commun.*, vol. 284, nos. 16–17, pp. 3895–3903, 2011.

[10] Y. Wu, "Image encryption using the two-dimensional logistic chaotic map," *J. Electron. Imag.*, vol. 21, no. 1, p. 013014, 2012.

[11] S. Lian, J. Sun, and Z. Wang, "A block cipher based on a suitable use of the chaotic standard map," *Chaos, Solitons Fractals*, vol. 26, no. 1, pp. 117–129, 2005.

[12] P. Ping, F. Xu, Y. Mao, and Z. Wang, "Designing permutation–substitution image encryption networks with Henon map," *Neurocomputing*, vol. 283, pp. 53–63, Mar. 2018.

[13] B. Norouzi, S. M. Seyedzadeh, S. Mirzakuchaki, and M. R. Mosavi, "A novel image encryption based on row-column, masking and main diffusion processes with hyper chaos," *Multimedia Tools Appl.*, vol. 74, no. 3, pp. 781–811, 2015.

[14] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[15] 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.

[16] H. Liu and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Comput. Math. Appl.*, vol. 59, no. 10, pp. 3320–3327, 2010.

[17] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Appl. Soft. Comput.*, vol. 12, no. 5, pp. 1457–1466, 2012.

[18] X. Huang and G. Ye, "An image encryption algorithm based on hyper-chaos and DNA sequence," *Multimedia Tools Appl.*, vol. 72, no. 1, pp. 57–70, 2014.

[19] X.-Y. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dyn.*, vol. 62, no. 3, pp. 615–621, 2010.

[20] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Process.*, vol. 148, pp. 124–144, Jul. 2018.

[21] X. Wang, L. Liu, and Y. Zhang, "A novel chaotic block image encryption algorithm based on dynamic random growth technique," *Opt. Lasers Eng.*, vol. 66, pp. 10–18, Mar. 2015.

[22] Z. Hua, S. Yi, and Y. Zhou, "Medical image encryption using high-speed scrambling and pixel adaptive diffusion," *Signal Process.*, vol. 144, pp. 134–144, Mar. 2018.

[23] L. Wang, H. Song, and P. Liu, "A novel hybrid color image encryption algorithm using two complex chaotic systems," *Opt. Lasers Eng.*, vol. 77, pp. 118–125, Feb. 2016.

[24] Z. Parvin, H. Seyedarabi, and M. Shamsi, "A new secure and sensitive image encryption scheme based on new substitution with chaotic function," *Multimedia Tools Appl.*, vol. 75, no. 17, pp. 10631–10648, 2016.

[25] C. Li, D. Lin, and J. Lü, "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE Multimedia*, vol. 24, no. 3, pp. 64–71, Mar. 2017.

[26] C. Li and K.-T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Process.*, vol. 91, no. 4, pp. 949–954, 2011.

[27] X. Tao, K. Wong, and X. Liao, "Selective image encryption using a spatiotemporal chaotic system," *Chaos*, vol. 17, no. 2, p. 023115, 2007.

[28] L. Teng and X. Wang, "A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive," *Opt. Commun.*, vol. 285, no. 20, pp. 4048–4054, 2012.

[29] C. Fu, B.-B. Lin, Y.-S. Miao, X. Liu, and J.-J. Chen, "A novel chaos-based bit-level permutation scheme for digital image encryption," *Opt. Commun.*, vol. 284, no. 23, pp. 5415–5423, 2011.

[30] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inf. Sci.*, vol. 181, no. 6, pp. 1171–1186, 2011.

[31] W. Zhang, K.-W. Wong, H. Yu, and Z.-L. Zhu, "A symmetric color image encryption algorithm using the intrinsic features of bit distributions," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 3, pp. 584–600, 2013.

[32] J.-X. Chen, Z.-L. Zhu, C. Fu, L.-B. Zhang, and Y. Zhang, "An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach," *Commun. Nonlinear Sci. Numer Simul.*, vol. 23, nos. 1–3, pp. 294–310, Jun. 2015.

[33] W. Zhang, H. Yu, Y.-L. Zhao, and Z.-L. Zhu, "Image encryption based on three-dimensional bit matrix permutation," *Signal Process.*, vol. 118, pp. 36–50, Jan. 2016.

[34] W. Cao, Y. Zhou, C. L. P. Chen, and L. Xia, "Medical image encryption using edge maps," *Signal Process.*, vol. 132, pp. 96–109, Mar. 2017.

[35] G. Ye and X. Huang, "An efficient symmetric image encryption algorithm based on an intertwining logistic map," *Neurocomputing*, vol. 251, pp. 45–53, Aug. 2017.

[36] H.-M. Yuan, Y. Liu, T. Lin, T. Hu, and L.-H. Gong, "A new parallel image cryptosystem based on 5D hyper-chaotic system," *Signal Process., Image Commun.*, vol. 52, pp. 87–96, Mar. 2017.

[37] M. Hénon, "A two-dimensional mapping with a strange attractor," *Commun. Math. Phys.*, vol. 50, no. 1, pp. 69–77, 1976.

[38] P. Ping, Y. Mao, X. Lv, F. Xu, and G. Xu, "An image scrambling algorithm using discrete Henon map," in *Proc. IEEE Int. Conf. Inf. Automat.*, Aug. 2015, pp. 429–432.

[39] M. Mollaeefar, A. Sharif, and M. Nazari, "A novel encryption scheme for colored image based on high level chaotic maps," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 607–629, Jan. 2017.

[40] L. Y. Zhang, C. Li, K.-W. Wong, S. Shu, and G. Chen, "Cryptanalyzing a chaos-based image encryption algorithm using alternate structure," *J. Syst. Softw.*, vol. 85, no. 9, pp. 2077–2085, 2012.

[41] H. Zhu, X. Zhang, H. Yu, C. Zhao, and Z. Zhu, "A novel image encryption scheme using the composite discrete chaotic system," *Entropy*, vol. 18, no. 8, p. 276, 2016.

[42] A. Rukhin *et al.*, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication, Gaithersburg, MD, USA, Tech. Rep. 800-22 Rev 1a, 2010. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-22-revla/SP800-22revla.pdf

[43] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice," *Inf. Sci.*, vol. 273, pp. 329–351, Jul. 2014.

[44] N. K. Pareek, V. Patidar, and K. K. Sud, "Cryptography using multiple one-dimensional chaotic maps," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 10, no. 7, pp. 715–723, 2005.

[45] X. Wu, H. Kan, and J. Kurths, "A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps," *Appl. Soft Comput.*, vol. 37, pp. 24–39, Dec. 2015.

[46] Y. Wu, Y. Zhou, J. P. Noonan, and S. Agaian, "Design of image cipher using latin squares," *Inf. Sci.*, vol. 264, pp. 317–339, Apr. 2014.

[47] X. Liao, S. Lai, and Q. Zhou, "A novel image encryption algorithm based on self-adaptive wave transmission," *Signal Process.*, vol. 90, no. 9, pp. 2714–2722, 2010.

[48] Y.-Q. Zhang and X.-Y. Wang, "A new image encryption algorithm based on non-adjacent coupled map lattices," *Appl. Soft. Comput.*, vol. 26, pp. 10–20, Jan. 2015.

**PING PING** received the B.Sc. degree in communication engineering and the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2005 and 2009, respectively. She is currently an Associate Professor with the College of Computer and Information, Hohai University, Nanjing. Her research interests include data security and privacy, multimedia security, cloud computing security and many other aspects of cryptography.

**JINYANG FAN** is a graduate student with the College of Computer and Information, Hohai University, Nanjing, China. His research interests include data security and privacy, multimedia security.

**YINGCHI MAO** received the B.Sc. and M.Sc. degrees in computer science and technology from Hohai University in 1999 and 2003, respectively, and the Ph.D. degree in computer science and technology from Nanjing University, China, in 2007. She is currently an Associate Professor with the College of Computer and Information, Hohai University, Nanjing, China. Her research areas include distributed computing, wireless sensor networks, and distributed data management.

**FENG XU** is currently a Professor, the President of the Hohai University Young Scientists & Engineers Association, the Deputy Secretary General of Water Resources Informatization technique committee of the Chinese Hydraulic Engineering Society, the Deputy Secretary General of the Jiangsu Computer Society, and the Committee Member of Systems Software technique committee of the China Computer Federation. He has participated in several state and provincial projects and published a certain number of papers in refereed international journals and conference proceedings, also applied for some Chinese patents and software licenses. His research interests include cloud computing, network information security, and domain software engineering.

**JERRY GAO** has over 15 years of academic research and teaching experience and over 10 years of industry working and management experience on software engineering and IT development applications. He is currently a Professor with the Department of Computer Engineering, San Jose State University. He has published over hundreds of publications in IEEE/ACM journals, magazines, International conferences and workshops. He has co-authored three published technical books and edited numerous books in software engineering, software testing, and mobile computing. His current research areas include cloud computing, TaaS, software engineering, test automation, mobile computing, and cloud services.