Digital Signal Processing 93 (2019) 115–127

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Cryptanalyzing an image encryption based on a modified Henon map using hybrid chaotic shift transform

Kanglei Zhou, Minghui Xu, Jidong Luo, Haiju Fan, Ming Li *

College of Computer and Information Engineering, Henan Normal University, Xinxiang, 453007, China

ARTICLE INFO

Article history:
Available online 25 July 2019

Keywords:
Cryptanalysis
Chosen-plaintext attack
Modified Henon map
Diffusion

ABSTRACT

Recently, a novel image encryption scheme has been proposed based on a modified Henon map using hybrid chaotic shift transform. This paper analyzes the security of the original encryption scheme and finds it insecure against the chosen-plaintext attack. Meanwhile, an efficient strategy is proposed to break the original encryption scheme with several chosen-plaintext attacks. The experimental results show that all the keys can be revealed with a time complexity of only $O(\lceil MN\log_c(MN)\rceil)$. Furthermore, some improvement suggestions are proposed.

1. Introduction

With the rapid development of digital information technology and network technology, digital images are increasingly becoming an irreplaceable form of information acquisition for daily communication. Digital images carry a lot of confidential information due to their inherent characteristics, such as large capacity, high redundancy, and interpixel correlation [1,2]; the information in digital images is very different from traditional text information and is vulnerable to safety hazards. Therefore, secure storage and transmission of digital images have become the primary concern in multimedia communication. Image encryption is an effective method to prevent unauthorized access, such as interception, tampering, illegal copying and dissemination [3,4]. However, traditional encryption techniques including DES, AES and RSA expose many limitations to digital images with the characteristics of large storage capacity, high redundancy and strong correlation among adjacent pixels. To solve these problems, chaotic image encryption schemes have been proposed [5–8]. The encryption schemes based on chaos have certain properties, such as sensitivity to initial conditions and control parameters [9], complexity of computing power [10] and pseudo-randomness; these properties have led many researchers to propose secure and efficient encryption schemes based on chaos.

So far, many chaotic-based encryption schemes have been proposed. In [11], a novel cryptosystem based on transformed logistic maps was proposed using Fridrich's encryption structure [12] in which six odd secret keys and three chaotic keys are involved. A previous study [13] proved that the scheme was insecure against the chosen-plaintext attack (CPA). Nevertheless, most encryption schemes based on Fridrich's permutation-diffusion model [14] demonstrated that 96.7% of bit values were unaltered [15,16]. Unlike Fridrich's encryption structure, one round modified permutation-diffusion architecture [17] was based on the bit level rather than the pixel level, and the diffusion stage was based on the output of the classical affine cipher rather than plain pixels. The scheme operated on the bit level can be used to reduce redundancy and statistical links, so a variety of encryption schemes based on the bit level [18–21] have been proposed. However, this kind of scheme also has such shortcomings because there are repeated patterns in the permutation phase [21], and the operation requires considerable computation time [22]. The encryption scheme in [17] was broken successfully by Liu et al. in [23]. To improve the permutation performance, Wong et al. [24] proposed using an "add-and-then-shift" strategy by including some diffusion effects in the permutation phase where the iteration round and computational complexity can be reduced dramatically without affecting the security level of the synthetic cryptosystem. In [25], a circular inter-intra-pixels bit-level permutation-confusion strategy was proposed that not only takes advantage of this approach in Fridrich's design but also applies a confusion strategy to reduce the redundancy significantly. Bit-level circular shifting of each row has a repeated pattern and uniform bit distribution [26] that can decrease the correlation between adjacent high-level element planes.

In contrast, cryptanalysis is used to check whether the existing encryption mechanisms are practical or not. Many cryptanalytic methods [27,29,31,33] have been proposed to find out the security defects of the existing cryptosystems [28,30,32,34]. For instance, Li et al. in [27] applied CPA to break Wang's encryption scheme [28]. Li et al. in [29] successfully obtained the equivalent secret key of a

* Corresponding author.
E-mail address: liming@htu.edu.cn (M. Li).

https://doi.org/10.1016/j.dsp.2019.07.013
1051-2004/© 2019 Elsevier Inc. All rights reserved.

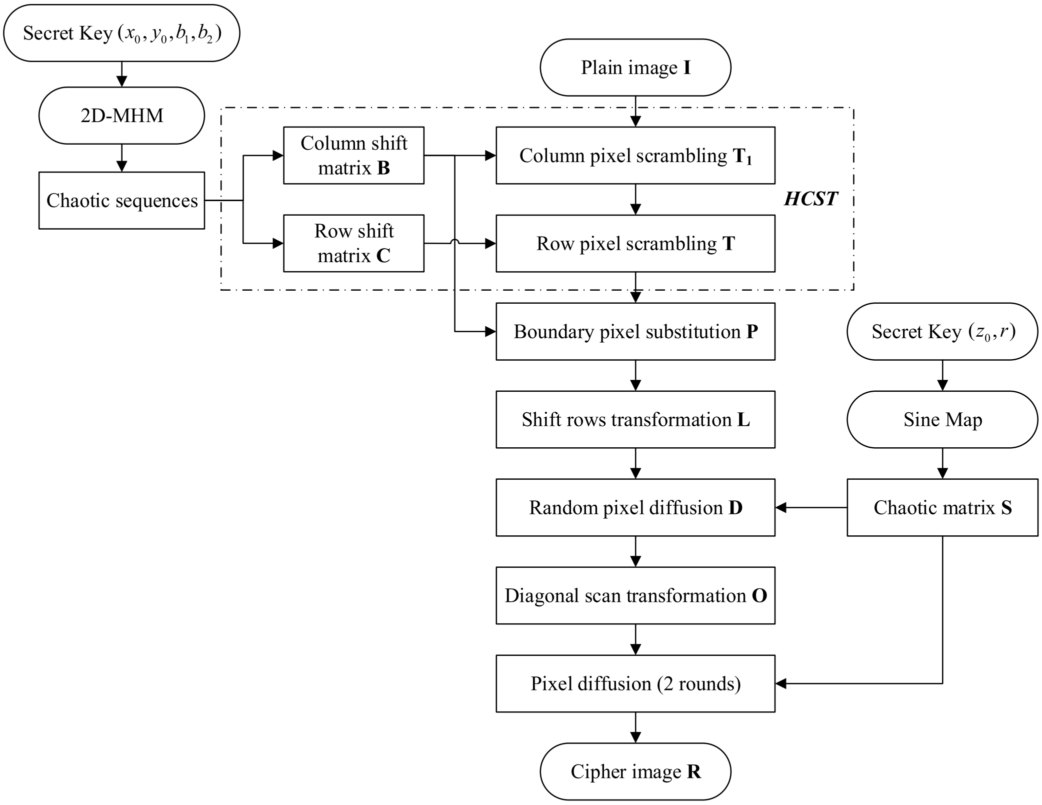

Secret Key (x_0, y_0, b_1, b_2)
2D-MHM
Chaotic sequences
Column shift matrix **B**
Row shift matrix **C**
Plain image **I**
Column pixel scrambling $\mathbf{T_1}$
Row pixel scrambling **T**
HCST
Boundary pixel substitution **P**
Secret Key (z_0, r)
Sine Map
Shift rows transformation **L**
Random pixel diffusion **D**
Chaotic matrix **S**
Diagonal scan transformation **O**
Pixel diffusion (2 rounds)
Cipher image **R**

Fig. 1. Flowchart of the original image encryption scheme.

chaotic image encryption algorithm based on information entropy [30] through a differential attack. Fan et al. in [31] broke the encryption scheme in [32] by combining the ciphertext-only attack and CPA. Zhang et al. in [33] applied CPA to break the encryption scheme based on a hyperchaotic system [34] with only one round diffusion process. The common problem of these compromised encryption schemes [28,30,32,34] is that the key stream is consistent. To overcome this shortcoming, different plaintext-related encryption schemes [35–37] have been proposed recently, in which the chaotic secret sequences are generated from both the secret key and the plain image at the same time.

In [38], a novel cryptosystem was proposed for encryption of gray images based on a modified Henon map using hybrid chaotic shift transform in which the chaotic sequences generated by a modified Henon map and sine map are applied to control the confusion through shift transform and the diffusion by the XOR operation. The scheme mainly includes two parts: the confusion process is controlled effectively by hybrid shift transform to change the pixels of the row and column, and the diffusion process is restrained availably by a chaotic matrix generated from a sine map. It is claimed that through the confusion and diffusion, images can be safely and effectively encrypted without being attacked. However, we found the original scheme was insecure against CPA. In this paper, we proposed a strategy to break it successfully. The major novelty of our cryptanalysis work is to eliminate the effect of the two-level diffusion by applying the permutation satisfying the distributive property with respect to XOR, which enriches the cryptanalysis research in terms of transforming the encryption structure.

This paper is organized in the following way. Section 2 briefly reviews the original encryption scheme. Section 3 demonstrates the detailed cryptanalysis method. The final section provides concluding remarks.

2. Review of the original image encryption scheme

The original image encryption scheme is shown in Fig. 1, which is a novel image cryptosystem based on a modified Henon map (2D-MHM) and a sine map using hybrid chaotic shift transform (HCST). HCST is used to perform a confusion operation controlled by 2D-MHM, and the principle of diffusion by the XOR operation is achieved by using a chaotic matrix generated by a sine map.

As shown in Fig. 1, a plain image is encrypted through six phases: hybrid chaotic shift transform, boundary pixels substitution, shift rows transformation, the first level of diffusion by the XOR operation, diagonal scanning transformation and the second level of diffusion by the XOR operation.

2.1. Hybrid chaotic shift transform

Suppose the size of the image is $M \times N$. Let $\mathbf{I} = \{I(i, j)\}_{i=1, j=1}^{M,N}$ and $\mathbf{T} = \{T(i, j)\}_{i=1, j=1}^{M,N}$ be the original image and its corresponding shuffled image, respectively. Then, the process of HCST can be defined as

$$\mathbf{T}_1 = F_1(\mathbf{I}, \mathbf{B}), \tag{1}$$

$$\mathbf{T} = F_2(\mathbf{T_1}, \mathbf{C}), \tag{2}$$

where $F_1(\cdot)$ and $F_2(\cdot)$ denote the cyclic column shift transformation and the cyclic row shift transformation, respectively; $\mathbf{B} = \{b_i\}_{i=1}^{N}$ and $\mathbf{C} = \{b_i\}_{i=1}^{M}$ are chaotic series matrices; b_i and c_i represent the step size of the cyclic up or down shift in column i and the step size of the cyclic right or left shift in row i, respectively. Moreover, the odd number of rows and columns are moved to the left and up, respectively, while the even numbers of rows and columns are moved in the opposite direction.

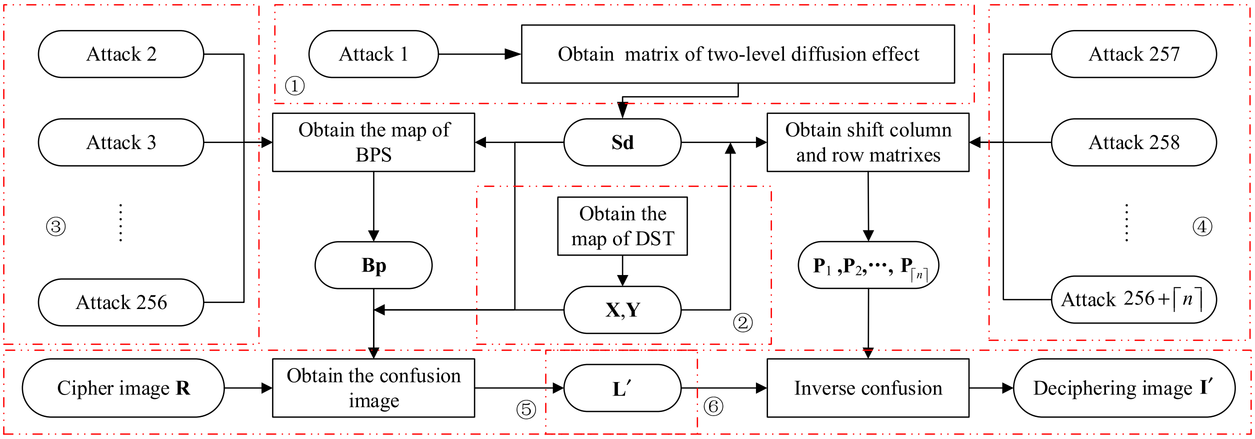


Fig. 2. Flowchart of the proposed attack strategy.

2.2. Boundary pixels substitution and shift rows transformation

For the shuffled image **T**, the boundary pixel value $T(i,j)$ of the first or last row is replaced by the value $b_{T(i,j)+1}$ of the column shift matrix **B**. The boundary pixels substitution results in the image $\mathbf{P}=\{P(i,j)\}_{i=1,j=1}^{M,N}$.

$$P(i,j)=\begin{cases} b_{T(i,j)+1}, i\in\{1,M\} \\ T(i,j), i\in(1,M) \end{cases} 1\leq j\leq N. \tag{3}$$

Additionally, shift rows transformation is applied to complicate dependence of the statistics of the ciphertext image on the plaintext image through this scheme, which operates one row at one time by shifting the byte to the left. Let $\mathbf{L}=\{L(i,j)\}_{i=1,j=1}^{M,N}$ be the result of shift rows transformation, which can be mathematically represented as

$$L(i,j)=\begin{cases} P(i,j+i-1), j-i+1\leq N \\ P(i,\mathrm{mod}(j+i-1,N)), j-i+1>N. \end{cases} \tag{4}$$

2.3. First level of diffusion

The diffusion process is divided into two levels and used to illustrate the effects of change in one bit of the plaintext on each bit of the ciphertext, which hides the statistical structure of the plaintext [40]. For the first level of diffusion, pixel values are changed randomly by the XOR operation with the chaotic matrix $\mathbf{S}=\{S(i,j)\}_{i=1,j=1}^{M,N}$ using Eq. (5).

$$\begin{aligned} &D(i,j)=L(i,j)\oplus S(i-1,N)\oplus S(i,j+1)1\leq i\leq M, \\ &\quad 1\leq j\leq N, \end{aligned} \tag{5}$$

where $\oplus$ denotes the bitwise XOR operation, **S** is generated from the sine map, and $\mathbf{D}=\{D(i,j)\}_{i=1,j=1}^{M,N}$ indicates the image after the first level diffusion operation.

2.4. Diagonal scanning transformation

Diagonal scanning transformation starts in the upper left hand corner and ends at the lower right hand corner. At the beginning of the transformation, the 2D image **D** is converted into a one-dimension (1D) array of size $1\times MN$ by scanning the image diagonally. Next, the 1D array is converted into the 2D image $\mathbf{O}=\{O(i,j)\}_{i=1,j=1}^{M,N}$.

2.5. Second level of diffusion

In this second level of the diffusion stage, the pixel values are changed by the XOR operation for each digit with the previous cipher digit, current permutated pixels and the chaotic matrix **S**. After applying two rounds of the diffusion operation with the XOR operation, the encrypted image $\mathbf{R}=\{R(i,j)\}_{i=1,j=1}^{M,N}$ can be obtained through Eq. (6).

$$R(i,j)=\begin{cases} O(i,j)\oplus S(i,j), i=1 \\ O(i,j)\oplus O(i-1,j)\oplus S(i,j), 1<i\leq M \end{cases} 1\leq j\leq N. \tag{6}$$

3. Cryptanalysis

To completely break the original encryption scheme, we need at most $256+\lceil n\rceil$ chosen plaintext images. A flowchart of our proposed attack strategy is shown in Fig. 2, in which "Attack i" represents the ith CPA. First, the equivalent matrix **Sd** for two-level diffusion is obtained by Attack 1. Second, since the diagonal scanning transformation is invertible, it is easy to get the mapping matrix of the transformations **X** and **Y**. Then, the equivalent matrix of the boundary pixels substitution **Bp** can be calculated by Attacks 2–256 with the help of **Sd**. After that, we can obtain shift matrices $\mathbf{P}_1,\mathbf{P}_2,\ldots,\mathbf{P}_{\lceil n\rceil}$ by Attacks 257–256 + $\lceil n\rceil$ and the known matrix **Sd**. Next, the diffusion effect of the cipher image **R** can be eliminated by the known matrix **Bp** and **Sd**. Eventually, the permutation-only image $\mathbf{L}'$ is recovered to the plain image $\mathbf{I}'$ with the help of the shift matrices $\mathbf{P}_1,\mathbf{P}_2,\ldots,\mathbf{P}_{\lceil n\rceil}$.

3.1. Eliminating the two-level diffusion effect

In our cryptanalysis work, we must solve the problem of the two-level diffusion first. Although there is diagonal scanning transformation between the two levels of diffusion, it is still easily to be broken using CPA. The permutated image **L** diffuses through two stages to become encrypted image **R**. It also goes through the diagonal scanning transformation between the two levels of diffusion. To decrypt the targeted cipher image so that we can obtain the permutated image, the first thing we must do is to eliminate the two-level diffusion effect by Attack 1 in Fig. 2.

In our paper, the permutation process we studied only involves one-to-one mapping of pixel positions, which can be mathematically expressed as follows:

$$\mathbf{R}=F(\mathbf{I}), \tag{7}$$

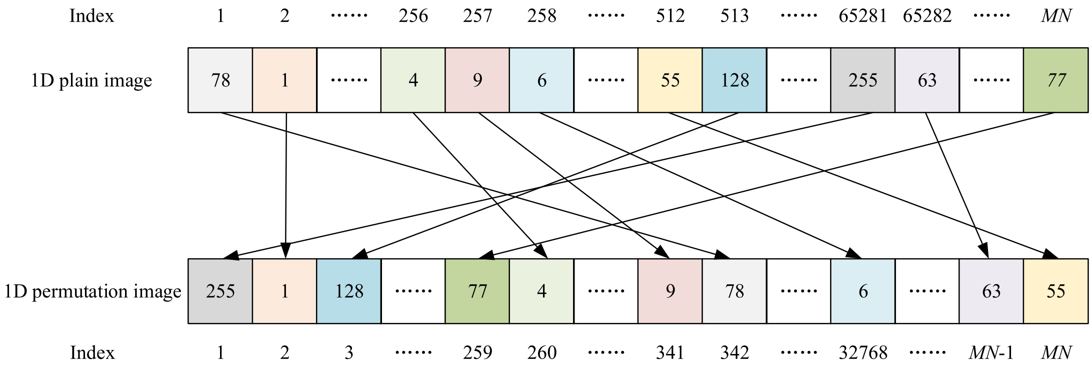


Fig. 3. 1D image permutation schematic.

where $F(\cdot)$ denotes the permutation transformation, and each pixel $I(i, j)$ in the plain image $\mathbf{I}$ uniquely corresponds to a specific pixel $R(p, q)$ in the permutated image $\mathbf{R}$. And the diffusion process we studied only involves the XOR operation, which can be mathematically defined as follows:

$$\mathbf{D} = \mathbf{I} \oplus \mathbf{S}, \tag{8}$$

where $\mathbf{S}$ and $\mathbf{D}$ are the diffusion matrix and the diffused image, respectively.

The following propositions contribute to the elimination of the two-level diffusion effect and calculation of **Sd**. Note that the following propositions apply only when the permutation process is implemented by changing the pixel positions and the diffusion process is performed by the XOR operation.

Proposition 1. *For the diffusion-permutation structure used in the encryption scheme under study, the order of the diffusion process and the permutation process is equivalently commutative. Properly speaking, the permutation satisfies the distributive property formally with respect to the XOR.*

For plain image $\mathbf{I}$ *of size* $M \times N$*, if the diffusion matrix is* $\mathbf{S}$*, then*

$$F(\mathbf{I} \oplus \mathbf{S}) = F(\mathbf{I}) \oplus F(\mathbf{S}). \tag{9}$$

Proof. For the plain image $\mathbf{I}$, pull it row by row into a 1D image $\mathbf{V}_I$ of size $MN \times 1$ with a permutation matrix $\mathbf{P}$ of size $MN \times MN$, and pull the diffusion matrix $\mathbf{S}$ of size $M \times N$ row by row into a 1D image $\mathbf{V}_S$ of size $MN \times 1$. Then, to prove the establishment of Eq. (9), we need to prove Eq. (10) first.

$$(\mathbf{V}_I \oplus \mathbf{V}_S) \times \mathbf{P} = (\mathbf{V}_I \times \mathbf{P}) \oplus (\mathbf{V}_S \times \mathbf{P}), \tag{10}$$

where the permutation matrix $\mathbf{P}$ is an identity matrix that swaps the order of rows, and it satisfies

$$\sum_{j=1}^{MN} P(i, j) = 1, 1 \leq i \leq MN, \tag{11}$$

$$\sum_{i=1}^{MN} P(i, j) = 1, 1 \leq j \leq MN. \tag{12}$$

Suppose the 1D permutation-only image is $\mathbf{V}_P$ of size $MN \times MN$, then

$$\mathbf{V}_P = \mathbf{P} \times \mathbf{V}_I, \tag{13}$$

where if $P(i, j) = 1$, there is a permutation mapping from i to $MN \times (i-1) + j$ between the 1D plain image and the 1D permutated image. Fig. 3 clearly shows the process for 1D image permutation.

Since matrix multiplication satisfies the distributive property of the XOR operation, it is obvious that Eq. (9) is true. Therefore, Proposition 1 has been proved.

Diagonal scanning transformation is a special permutation transform, and it satisfies the distributive property with respect to the XOR. An example in Fig. 4 is provided additionally to verify Proposition 1. Fig. 4(a) shows the process of diffusion by the XOR operation and then permutation, while Fig. 4(b) shows the process of permutation and then diffusion by the XOR operation.

In Fig. 4, assuming that there is an image of size 2 × 4, and the size of the diagonal scanning transformation matrix $\mathbf{P}$ is 8 × 8. It can be seen that the cipher images $\mathbf{C}$ in Fig. 4(a) and Fig. 4(b) respectively are the same, indicating that changing the order of permutation and diffusion would not affect the encryption result.

Therefore, the diagonal scanning transformation satisfies the distributive property with respect to the XOR, and the order of the diagonal scanning transformation and diffusion by the XOR operation in the original encryption scheme can be equivalently exchanged. □

Proposition 2. *For the encryption scheme with a structure of permutation-diffusion or diffusion-permutation where the diffusion is only for the XOR operation, full zero images can be used as chosen plaintext images to eliminate the diffusion effect. More precisely, the ciphertext image of the full zero image is equivalent to the original or the permuted diffusion matrix.*

Proof. For an image encryption scheme with a structure of permutation-diffusion or diffusion-permutation where the diffusion is only for the XOR operation, suppose the plain image is $\mathbf{I}$ of size $M \times N$, the diffusion matrix is $\mathbf{S}$, and the ciphertext image is $\mathbf{C}$.

1) For the permutation-diffusion structure under study, the permutation transformation is done first, and then the diffusion transformation is carried out. Then

$$\mathbf{C} = F(\mathbf{I}) \oplus \mathbf{S} = \mathbf{P} \oplus \mathbf{S}, \tag{14}$$

where $\mathbf{P}$ is the permutation-only image of $\mathbf{I}$.

If we take the full zero image $\mathbf{I}_0$ as the input image, the pixels of the full zero image $\mathbf{I}_0$ are not scrambled after the permutation phase, so the permutation-only image is still $\mathbf{I}_0$. After the diffusion stage, the ciphertext image $\mathbf{C}_0$ can be obtained.

$$\mathbf{C}_0 = F(\mathbf{I}_0) \oplus \mathbf{S} = \mathbf{I}_0 \oplus \mathbf{S} = \mathbf{S}. \tag{15}$$

Clearly, the obtained ciphertext image $\mathbf{C}_0$ is just equal to the diffusion matrix $\mathbf{S}$.

Thus, we can use the ciphertext image $\mathbf{C}_0$ to eliminate the diffusion effect by the XOR operation.

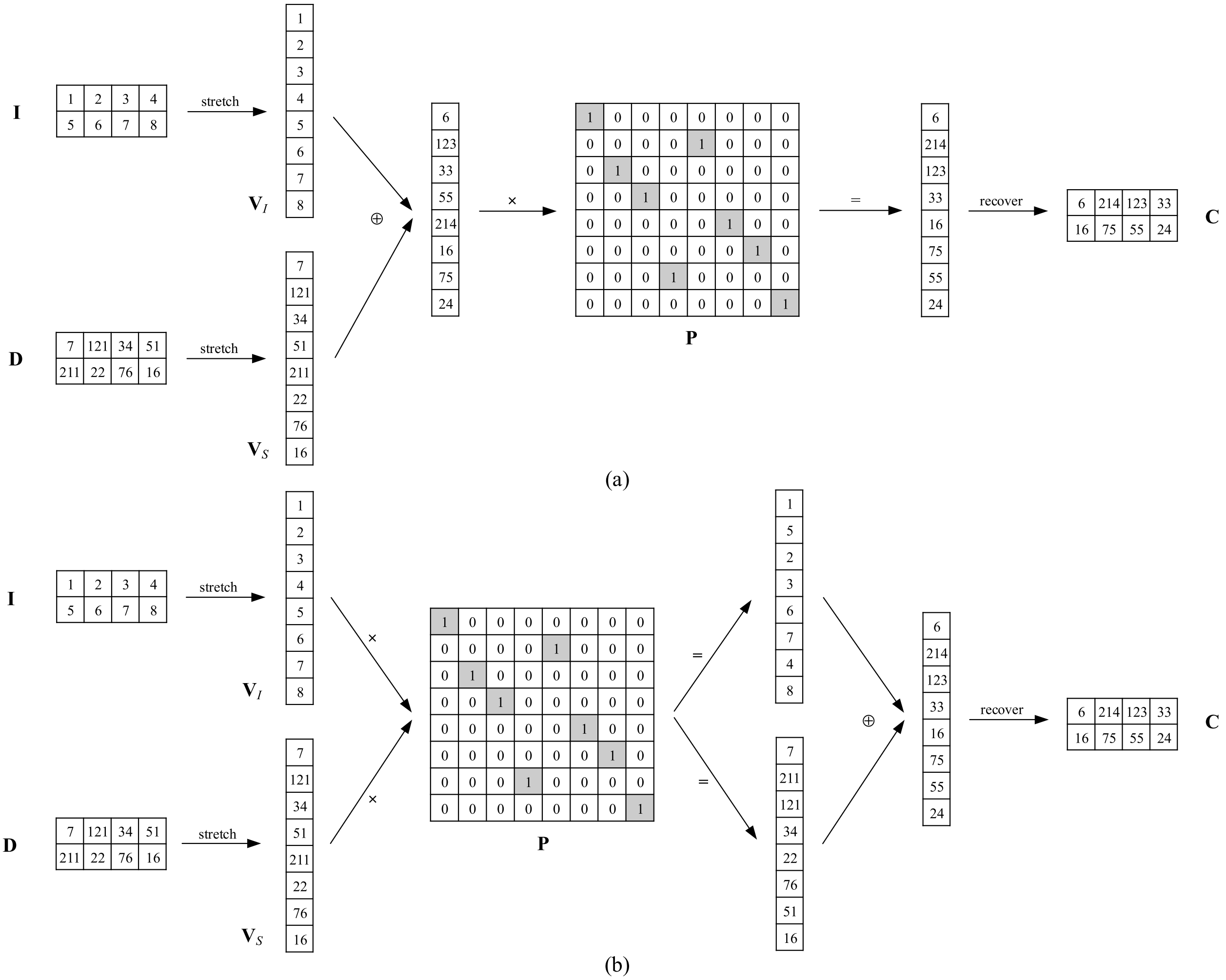


Fig. 4. Example of Proposition 1: (a) the process of diffusion by the XOR operation and then permutation; (b) the process of permutation and then diffusion by the XOR operation.

$$\mathbf{C} \oplus \mathbf{C}_0 = \mathbf{P} \oplus \mathbf{S} \oplus \mathbf{S} = \mathbf{P}. \tag{16}$$

2) For the diffusion-permutation structure under study, the diffusion transformation is done first, and then the permutation transformation is carried out. Then

$$\mathbf{C} = F(\mathbf{I} \oplus \mathbf{S}) = F(\mathbf{D}), \tag{17}$$

where **D** is the diffusion-only image of **I**.

We also take the full zero image as the input image. As shown in Eq. (18), the diffused image is just the diffusion matrix **S**. Therefore, the obtained ciphertext image $\mathbf{C}_0$ is exactly the permutated image of **S**.

$$\mathbf{C}_0 = F(\mathbf{I}_0 \oplus \mathbf{S}) = F(\mathbf{S}). \tag{18}$$

Based on Proposition 1, we can use the ciphertext image $\mathbf{C}_0$ to eliminate the diffusion effect by the XOR operation shown in Eq. (19).

$$\begin{aligned} \mathbf{C} \oplus \mathbf{C}_0 &= F(\mathbf{I} \oplus \mathbf{S}) \oplus F(\mathbf{S}) \\ &= F(\mathbf{I} \oplus \mathbf{S} \oplus \mathbf{S}) \\ &= F(\mathbf{I}) = \mathbf{P}. \qquad \square \end{aligned} \tag{19}$$

3.1.1. *Obtaining the matrix* **Sd**

The matrix **Sd** can be used to eliminate the diffusion effect of the ciphertext image, which is equivalent to the ciphertext image of the full zero image and is only determined by the diffusion matrix **S**. In other words, the ciphertext image of the full zero image can be used to eliminate the diffusion effect.

To get rid of the diffusion effect during the attack, **Sd** must be obtained. From Proposition 1, the cipher image **R** diffused by the second level can be obtained.

$$R(i,j) = \begin{cases} D'(i,j) \oplus S'(i,j) \oplus S(i,j), \quad i = 1 \\ D'(i,j) \oplus D'(i-1,j) \oplus S'(i,j) \oplus S'(i-1,j) \oplus S(i,j), \\ \quad i \neq 1, \end{cases} \tag{20}$$

where the image $\mathbf{D}' = \{D'(i,j)\}_{i=1,j=1}^{M,N}$ is the image **D** after the diagonal scanning transformation, and the matrix $\mathbf{S}' = \{S'(i,j)\}_{i=1,j=1}^{M,N}$ is the diffusion matrix **S** after the diagonal scanning transformation.

Meanwhile, it can be seen from Eq. (20) that the diffusion effect matrix **Sd** is only determined by the diffusion matrix **S**. For the full zero image, the permutation does not work, so the pixel values of $\mathbf{D}'$ are still all zero. Thus, the ciphertext image of the full zero image can be used to eliminate the diffusion effect, which is exactly equivalent to the matrix **Sd**.

Based on Proposition 2, the result of the cipher image for any plain image exclusive or the cipher image of the full zero image is equivalent to elimination of the two-level diffusion effect. In other words, let $\mathbf{R}_0 = \{R_0(i,j)\}_{i=1,j=1}^{M,N}$ be the encrypted image of the full

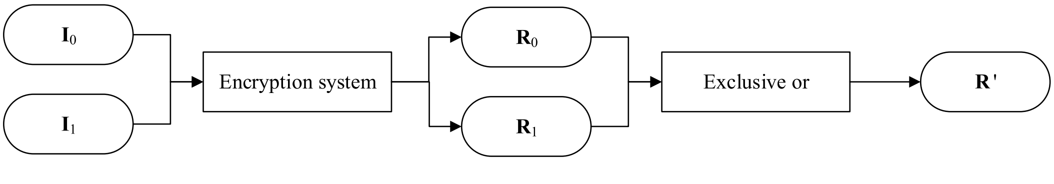
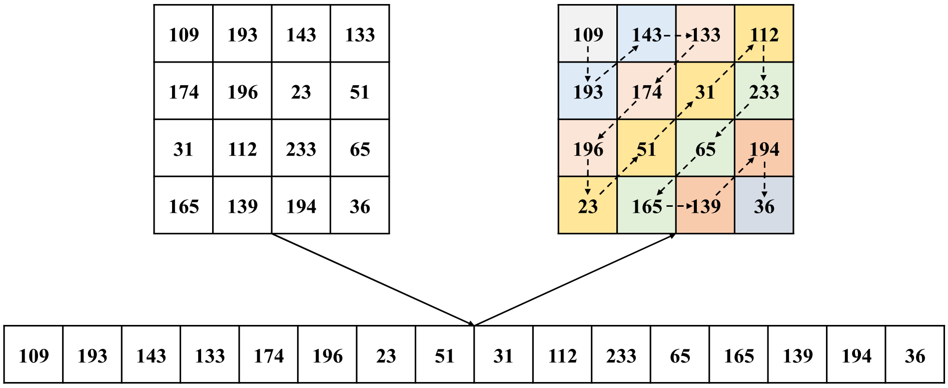


Fig. 5. Flowchart of elimination of the second level diffusion effect.



109	193	143	133
174	196	23	51
31	112	233	65
165	139	194	36

109	143	133	112
193	174	31	233
196	51	65	194
23	165	139	36

109	193	143	133	174	196	23	51	31	112	233	65	165	139	194	36

Fig. 6. Schematic diagram of the reverse diagonal scanning mechanism.

zero image, and $\mathbf{R}_1 = \{R_1(i,j)\}_{i=1,j=1}^{M,N}$ is the cipher image of any plain image. Then, $\mathbf{R}' = \{R'(i,j)\}_{i=1,j=1}^{M,N}$ is an image without the two-level diffusion effect if it is the result of $\mathbf{R}_0$ exclusive or $\mathbf{R}_1$.

Thus, we can obtain

$$\mathbf{Sd} = \mathbf{R}_0, \tag{21}$$

where $\mathbf{R}_0$ is the ciphertext image of the full zero image.

3.1.2. Eliminating the first level diffusion effect

A flowchart of elimination of the second level diffusion effect is shown in Fig. 5, where two chosen plain gray images $\mathbf{I}_0 = \{I_0(i,j)\}_{i=1,j=1}^{M,N}$ (Attack 1 in Fig. 2), which is full of zero pixels, and $\mathbf{I}_1 = \{I_1(i,j)\}_{i=1,j=1}^{M,N}$ (any Attack in Fig. 2), which can be any plain image, are used in an attack to eliminate the two-level diffusion effect.

Let the two cipher images be $\mathbf{R}_0 = \{R_0(i,j)\}_{i=1,j=1}^{M,N}$ and $\mathbf{R}_1 = \{R_1(i,j)\}_{i=1,j=1}^{M,N}$. Based on Proposition 2, $\mathbf{R}_0$ is the required matrix $\mathbf{Sd}$. Therefore, the result image $\mathbf{R}' = \{R'(i,j)\}_{i=1,j=1}^{M,N}$ of the XOR operation in Eq. (22) is an image in which the second level diffusion effect is eliminated.

$$R'(i,j) = R_0(i,j) \oplus R_1(i,j). \tag{22}$$

From Eq. (6), it is known that the cipher image $\mathbf{R}_0$ and $\mathbf{R}_1$ satisfy

$$R_0(i,j) = \begin{cases} O_0(i,j) \oplus S(i,j), & i = 1 \\ O_0(i,j) \oplus O_0(i-1,j) \oplus S(i,j), & i \neq 1, \end{cases} \tag{23}$$

$$R_1(i,j) = \begin{cases} O_1(i,j) \oplus S(i,j), & i = 1 \\ O_1(i,j) \oplus O_1(i-1,j) \oplus S(i,j), & i \neq 1, \end{cases} \tag{24}$$

where $\mathbf{O}_0 = \{O_0(i,j)\}_{i=1,j=1}^{M,N}$ and $\mathbf{O}_1 = \{O_1(i,j)\}_{i=1,j=1}^{M,N}$ are the images obtained after the diagonal scanning transformation.

Thus, $\mathbf{R}'$ can also be described as

$$R'(i,j) = \begin{cases} O_0(i,j) \oplus O_1(i,j), \quad i = 1 \\ O_0(i,j) \oplus O_1(i,j) \oplus O_0(i-1,j) \oplus O_1(i-1,j), \\ \quad i \neq 1. \end{cases} \tag{25}$$

Meanwhile, to eliminate the second level diffusion effect by the XOR operation thoroughly, we know from Eq. (6) that it also needs to pass through the row transformation by the XOR operation as

$$O'(i,j) = \begin{cases} R'(i,j), & i = 1 \\ R'(i,j) \oplus R'(i-1,j), & 1 < i \leq M \end{cases}, 1 \leq j \leq N, \tag{26}$$

where $\mathbf{O}' = \{O'(i,j)\}_{i=1,j=1}^{M,N}$ is the resulting cipher image of the elimination of the second level diffusion effect.

From Eq. (25), the cipher image $\mathbf{O}'$ can also be represented as

$$O'(i,j) = O_0(i,j) \oplus O_1(i,j) \quad 1 \leq i \leq M, 1 \leq j \leq N. \tag{27}$$

3.1.3. Obtaining the map of the diagonal scanning transformation

It is seen from the diagonal scanning transformation process of Section 2.4 that the transformation is reversible. A schematic diagram of a reverse diagonal scanning mechanism is shown in Fig. 6, which shows precisely that the pixel values are shuffled sufficiently. The 2D matrix is first transformed into a 1D row vector. Then, the 1D row vector is converted into a 2D image.

How the map of diagonal scanning transformation is obtained is described in Algorithm 1, which simulates the diagonal scanning transformation process. The input parameters M and N are the size of the plain image, and the output parameters are the map of the diagonal scanning transformation $\mathbf{X} = \{x_i\}_{i=1}^{MN}$ and $\mathbf{Y} = \{y_i\}_{i=1}^{MN}$. Suppose the images before and after the transformation are converted into a 1D array, (i,j) is the coordinate of the original image and (x_i, y_j) is the coordinate after the diagonal scanning transformation.

3.1.4. Inverse diagonal scanning transformation

A flowchart of the inverse diagonal scanning transformation is shown in Fig. 7. For the inverse diagonal scanning transformation, it is known that the distributive law is satisfied by the operation of an exclusive or an inverse diagonal scanning transformation based on Proposition 1. Therefore, the images $\mathbf{D}_0 = \{D_0(i,j)\}_{i=1,j=1}^{M,N}$ and $\mathbf{D}_1 = \{D_1(i,j)\}_{i=1,j=1}^{M,N}$ can be obtained by the inverse diagonal scanning transformation with the map $\mathbf{X}$ and $\mathbf{Y}$. Then, the image $\mathbf{D}' = \{D'(i,j)\}_{i=1,j=1}^{M,N}$can be obtained by the XOR operation between $\mathbf{D}_0$ and $\mathbf{D}_1$.

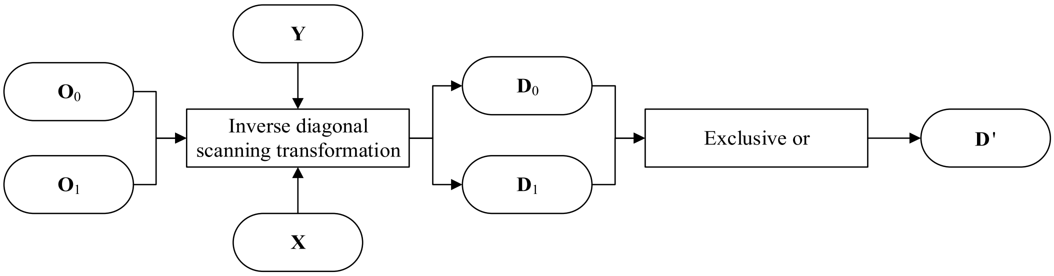


Fig. 7. Flowchart of the inverse diagonal scanning transformation.

Algorithm 1 Obtain the map of diagonal scanning transformation.

```
Input: The size of plain image $M$ and $N$
Output: The map X and Y of diagonal scanning transformation, which represent a map from $(i, j)$ to $(X_i, Y_j)$ after the diagonal scanning transformation.
procedure DST_map($M$, $N$)
    $i, j, X_1, Y_1, count \leftarrow 1$
    while $i \neq M$ and $j \neq N$ do
        $count \leftarrow count + 1$
        if $i + 1 \leq M$ then
            $X_{count} \leftarrow i + 1$
            $Y_{count} \leftarrow j$
            $i \leftarrow i + 1$
        else if $j + 1 \leq N$
                $X_{count} \leftarrow i$
                $Y_{count} \leftarrow j + 1$
                $j \leftarrow j + 1$
            end if
        end if
        while $i - 1 \geq 1$ and $j + 1 \leq N$ do
            $count \leftarrow count + 1$
            $X_{count} \leftarrow i - 1$
            $Y_{count} \leftarrow j + 1$
        end while
        $count \leftarrow count + 1$
        if $j + 1 \leq N$ then
            $X_{count} \leftarrow i$
            $Y_{count} \leftarrow j + 1$
            $j \leftarrow j + 1$
        else if $i + 1 \leq M$ then
                $X_{count} \leftarrow i + 1$
                $Y_{count} \leftarrow j$
                $j \leftarrow j + 1$
            end for
        end if
        while $i + 1 \leq M$ and $j - 1 \geq 1$ do
            $count \leftarrow count + 1$
            $X_{count} \leftarrow i + 1$
            $Y_{count} \leftarrow j - 1$
        end while
    end while
    return X, Y
end procedure
```

$$ind = (i-1) \times M + j, 1 \leq i \leq M, 1 \leq j \leq N, \tag{28}$$

$$D_0(x_{ind}, y_{ind}) = O_0(i, j), \tag{29}$$

$$D_1(x_{ind}, y_{ind}) = O_1(i, j), \tag{30}$$

where *ind* is the index of **X** and **Y**.

$$D'(i, j) = D_0(i, j) \oplus D_1(i, j) \quad 1 \leq i \leq M, 1 \leq j \leq N. \tag{31}$$

From Eq. (5) in the first level of diffusion by the XOR operation, it is obtained as

$$\begin{cases} D_0(i, j) = L_0(i, j) \oplus S(i-1, N) \oplus S(i, j+1) \\ D_1(i, j) = L_1(i, j) \oplus S(i-1, N) \oplus S(i, j+1) \\ 1 \leq i \leq M, 1 \leq j \leq N. \end{cases} \tag{32}$$

Thus,

$$D'(i, j) = L_0(i, j) \oplus L_1(i, j) \quad 1 \leq i \leq M, 1 \leq j \leq N. \tag{33}$$

In other words, $\mathbf{D}'$ has been removed from the first level of diffusion by the XOR operation. As a result, we can obtain

$$L'(i, j) = D'(i, j) \quad 1 \leq i \leq M, 1 \leq j \leq N, \tag{34}$$

where $\mathbf{L}' = \{L'(i, j)\}_{i=1,j=1}^{M,N}$ is the cipher image in which the two-level diffusion effect has been eliminated.

Let $\mathbf{I}_0 = \{I_0(i, j)\}_{i=1,j=1}^{M,N}$ be the full zero image. Because each row and column have a pixel value of 0, $\mathbf{I}_0$ is apparently unchanged during the process of HCST. Therefore, the result of HCST $\mathbf{T}_0 = \{T_0(i, j)\}_{i=1,j=1}^{M,N}$ can be expressed as

$$\mathbf{T}_1 = \mathbf{I}_0, \tag{35}$$

$$\mathbf{T}_0 = \mathbf{T}_1. \tag{36}$$

After the boundary pixels substitution, the pixel values for the first and last rows of $\mathbf{T}_0$ are replaced with the value of b_1. Therefore, we can obtain

$$\mathbf{P}_0 = \begin{bmatrix} b_1 & b_1 & b_1 & \cdots & b_1 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ b_1 & b_1 & b_1 & \cdots & b_1 \end{bmatrix}_{M \times N}, \tag{37}$$

where $\mathbf{P}_0 = \{P_0(i, j)\}_{i=1,j=1}^{M,N}$ is the result of the boundary pixels substitution, and all of its rows, except the first and last row, have a pixel value of 0.

Then, $\mathbf{L}_0$ can be obtained after shift rows transformation from Eq. (4)

$$\mathbf{L}_0 = \begin{bmatrix} b_1 & b_1 & b_1 & \cdots & b_1 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ b_1 & b_1 & b_1 & \cdots & b_1 \end{bmatrix}_{M \times N}. \tag{38}$$

Thus

$$L'(i, j) = \begin{cases} b_1 \oplus L_1(i, j), i \in \{1, M\} \\ L_1(i, j), i \in (1, M) \end{cases}. \tag{39}$$

To illustrate how to eliminate the two-level diffusion effect, for 1 chosen plaintext image $\mathbf{I}_0$, all of its pixels are given a value of 0, and its cipher image is shown in Fig. 8(a1). For the other chosen-plaintext image $\mathbf{I}_1 \in [0, 255]$ which is shown in Fig. 8(a2), we want to attack its cipher image, which is shown in Fig. 8(a3). The XOR result between them is shown in Fig. 8(b1). Then, the two results regarding the XOR operation of the row and the inverse diagonal scanning transformation are shown in Fig. 8(b2) and Fig. 8(b3), respectively. It is obvious that the two-level diffusion from the XOR operation is eliminated from the cipher image.

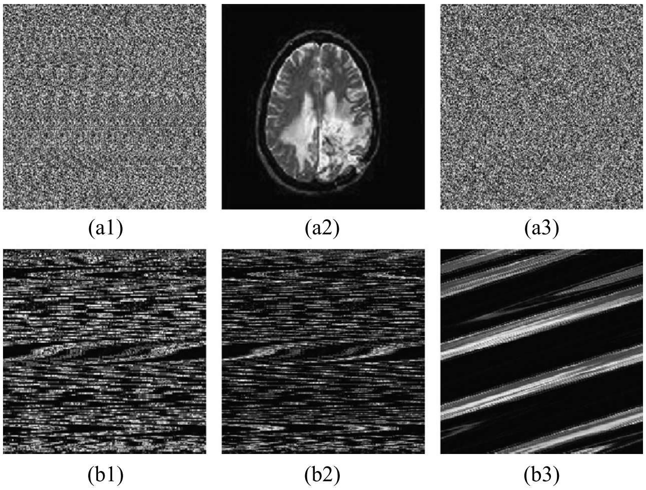
3.2. Reverse shift rows transformation

In the original encryption scheme, shift rows transformation is performed before the diffusion phase. Therefore, it is needed to perform the reverse shift rows transformation before the attack of boundary pixels substitution. Because the transformation rule is simple and does not require a key to participate, it is easy to break.

From Eq. (4), the image needs to be transformed by shifting rows in the encryption process. Therefore, the reverse shift rows transformation is required during the attack process. The reverse shift rows transformation moves the pixels of each row to the right by byte, which means row 1 no longer encounters any other shift and the last row is shifted by $M - 1$ bytes. This operation can be represented as

$$P'(i, j) = \begin{cases} L'(i, j-i+1), j-i+1 > 0 \\ L'(i, j-i+1+N), j-i+1 \leq 0 \end{cases}, 1 \leq j \leq N, \tag{40}$$

where $\mathbf{P}' = \{P'(i, j)\}_{i=1,j=1}^{M,N}$ is the result of reversing the shift rows transformation.


(a1) (a2) (a3)

(b1) (b2) (b3)

Fig. 8. Experiments of eliminating two-level diffusion. (a1) cipher image of the chosen plain image with all 0 pixels, (a2) the other chosen plain image, (a3) cipher image corresponding to (a2), (b1) cipher image corresponding to the results of (a1) and (a3) from the XOR operation, (b2) cipher image corresponding to (b1) after the XOR operation of the row, and (b3) cipher image corresponding to (b2) after inverse diagonal scanning transformation.

3.3. *Computing the boundary replacement matrix* **Bp**

After solving the shift rows transformation, we should obtain the boundary replacement matrix $\mathbf{Bp} = \{Bp_i\}_{i=1}^{N}$. The boundary replacement matrix determines the substitution sequence used in the boundary pixels substitution process where the first and last rows of the image are replaced with pixels. However, it is difficult to directly obtain the required replacement matrix **B** in the original encryption scheme, and we attempt to find the equivalent substitution matrix **Bp** by Attacks 2-256 instead.

A flowchart of the process for computing the boundary replacement matrix is shown in Fig. 9 from Eq. (39). For a cipher image after the elimination of the two-level diffusion effect, its first and last rows are computed with b_1 by the XOR operation. Suppose the pixel values of the image $\mathbf{I}_{all\_val}$ are all $val \in [1, 255]$. In the original encryption scheme, its image is not permutated in the process of confusion. However, the pixel values for its first and last rows are replaced with a value of b_{val+1} from Eq. (3). First, its cipher image $\mathbf{R}_{all\_val}$ is removed from the two-level diffusion effect. As a result, $\mathbf{L}'_{all\_val} = \{L'_{all\_val}(i,j)\}_{i=1,j=1}^{M,N}$ can be obtained.

$$L'_{all\_val}(i,j) = \begin{cases} b_1 \oplus b_{val+1}, i \in \{1, M\} \\ L_{all\_val}(i,j), i \in (1, M) \end{cases}. \tag{41}$$

From Eq. (41), each cipher image after the elimination of the two-level diffusion effect $\mathbf{L}'_{all\_val}$ has its first row and last row determined with **B**. Nevertheless, it is hard to figure out what **B** is. Therefore, we can easily figure **Bp** out, which is defined as

$$Bp_{val+1} = b_1 \oplus b_{val+1}, 1 \le val \le 255. \tag{42}$$

Because the original pixel value of *val* is replaced with Bp_{val+1} after the elimination of the two-level diffusion effect, the replaced pixel value of Bp_{val+1} should be substituted for *val* in the process of the attack.

3.4. *Obtaining the permutation equivalent mapping*

After the boundary replacement matrix **Bp** has been calculated, the cipher image can be restored to the permutation-only image **T**.

$$T(i,j) = \begin{cases} Bp_{P(i,j)}, i \in \{1, M\} \\ P(i,j), i \in (1, M) \end{cases}, 1 \le j \le N. \tag{43}$$

Therefore, the latest cryptanalysis results [39] for permutation-only image encryption schemes can be used to obtain the permutation equivalent mapping. From *Lemma* 1 used in [39], the permutation equivalent mapping can be acquired by n pairs of plain/cipher images.

$$n \ge \lceil \log_c(MN) \rceil, \tag{44}$$

where c is the number of assigned values in the locations or different color intensities, MN denotes the number of locations, and operation $\lceil x \rceil$ rounds variable x to the nearest integer toward infinity. In this paper, $c = 256$ and the number n of gray images can be calculated by Eq. (45).

$$n = \lceil \log_{256}(MN) \rceil. \tag{45}$$

For a plain gray image $\mathbf{I} = \{I(i,j)\}_{i=1,j=1}^{M,N}$, elements of the location vector can be placed through $\boldsymbol{p} = \{p_i\}_{t=1}^{MN} = [0, 1, 2, \cdots, MN - 1]$. Furthermore, the number of chosen plain gray images to attack is also n.

1) For instance, if the size of a plain gray image is 256 × 256, the number $n = 2$ and the number of chosen plain images is 2. First, each element of the vector $\boldsymbol{p} = \{p_i\}_{t=1}^{256\times256} = [0, 1, 2, \cdots, 256 \times 256 - 1]$ can be arranged row by row in matrix $\mathbf{A}_1$ of 256×256, of which each element includes 2 digits in base 256 across the character set $[0, 1, 2, \cdots, 255]$ to express $[0, 1, 2, \cdots, 256 \times 256 - 1]$.

$$\mathbf{A}_1 = \begin{bmatrix} (0)(0) & (0)(1) & (0)(2) & \cdots & (0)(255) \\ (1)(0) & (1)(1) & (1)(2) & \cdots & (1)(255) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (254)(0) & (254)(1) & (254)(2) & \cdots & (254)(255) \\ (255)(0) & (255)(1) & (255)(2) & \cdots & (255)(255) \end{bmatrix}_{256\times256}, \tag{46}$$

where the element $(q_1)(q_2)$ is corresponding to $256 \times q_1 + q_2$ of $\boldsymbol{p}$.

Next, two matrices with entries $0, 1, 2, \cdots, 255$ are obtained by splitting matrix $\mathbf{A}_1$ into two bit-plane images that should be the chosen gray plain images. If the chosen gray plain images are supposed to be $\mathbf{I}_{256} = \{I_{256}(i,j)\}_{i=1,j=1}^{M,N}$ (Attack 256 in Fig. 2) and

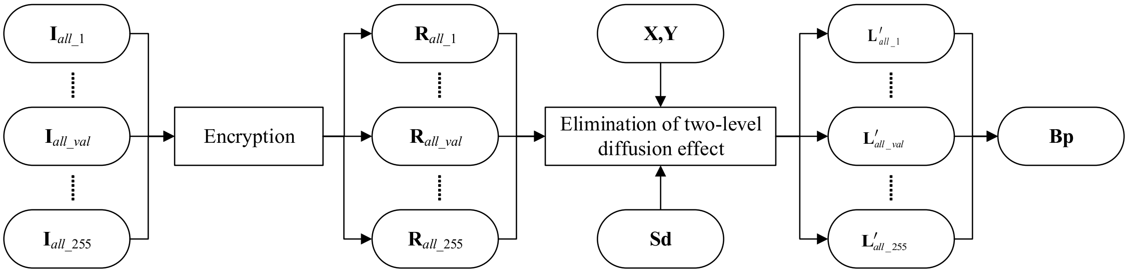


Fig. 9. Flowchart of the process for computing the boundary replacement matrix.

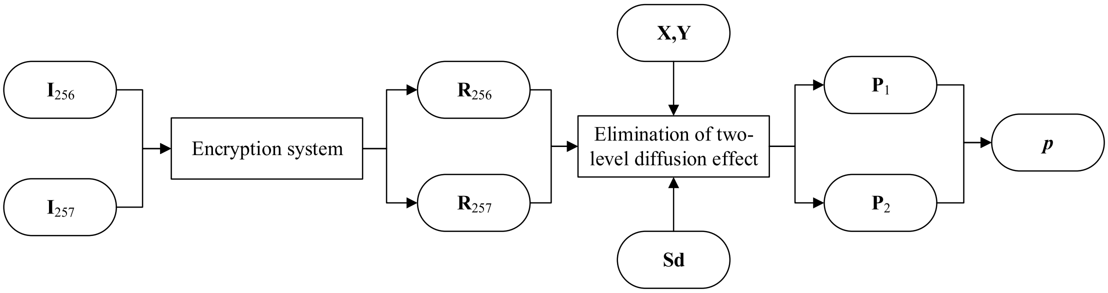


Fig. 10. Flowchart for obtaining the permutation rule.

$\mathbf{I}_{257} = \{I_{257}(i,j)\}_{i=1,j=1}^{M,N}$ (Attacks 257 in Fig. 2), they can be defined as

$$\mathbf{I}_{256} = \begin{bmatrix} 0 & 1 & 2 & \cdots & 255 \\ 0 & 1 & 2 & \cdots & 255 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 2 & \cdots & 255 \\ 0 & 1 & 2 & \cdots & 255 \end{bmatrix}_{256\times256}, \tag{47}$$

$$\mathbf{I}_{257} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 254 & 254 & 254 & \cdots & 254 \\ 255 & 255 & 255 & \cdots & 255 \end{bmatrix}_{256\times256}. \tag{48}$$

A flowchart for obtaining the permutation equivalent mapping $\boldsymbol{p}$ is shown in Fig. 10.

Meanwhile, the plain image of $\mathbf{I}_{256}$ is denoted by $\mathbf{R}_{256} = \{R_{256}(i,j)\}_{i=1,j=1}^{M,N}$. Similarly, the plain image of $\mathbf{I}_{257}$ is denoted by $\mathbf{R}_{257} = \{R_{257}(i,j)\}_{i=1,j=1}^{M,N}$. The permutation equivalent mapping is obtained as follows:

Step 1 Eliminate the two-level diffusion effect with the known **Sd**

$$O_{256}(i,j) = Sd(i,j) \oplus R_{256}(i,j) \quad 1 \le i \le 256, 1 \le j \le 256. \tag{49}$$

$$D_{256}(i,j) = \begin{cases} O_{256}(i,j), i = 1 \\ O_{256}(i,j) \oplus O_{256}(i-1,j), 1 < i \le 256, \end{cases} \tag{50}$$
$1 \le j \le 256.$

Then, do the inverse diagonal scanning transformation as

$$\begin{cases} ind = (i-1) \times M + j \\ L(x_{ind}, y_{ind}) = D(i,j) \end{cases} 1 \le i \le 256, 1 \le j \le 256. \tag{51}$$

Next, reverse the shift rows transformation as

$$P_{256}(i,j) = \begin{cases} L_{256}(i, j+i-1), j+i-1 \le N \\ L_{256}(i, (j+i-1)\mathrm{mod}N), j+i-1 \le 0 \end{cases} \tag{52}$$

Analogously, $\mathbf{P}_{257}$ is obtained in the same way.

Step 2 Calculate the permutation-only images $\mathbf{T}_{255}$ and $\mathbf{T}_{256}$ with the known **Bp**.

$$T_{256}(i,j) = \begin{cases} Bp(1, P_{256}(i,j)), i \in \{1, 256\} \\ P_{256}(i,j), i \in (1, 256) \end{cases}, 1 \le j \le 256. \tag{53}$$

$$T_{257}(i,j) = \begin{cases} Bp(1, P_{257}(i,j)), i \in \{1, 256\} \\ P_{257}(i,j), i \in (1, 256) \end{cases}, 1 \le j \le 256. \tag{54}$$

Step 3 Obtain the permutation matrix $\mathbf{T}_p = \{T_p(i,j)\}_{i=1,j=1}^{256,256}$.

$$\mathbf{T}_p = 256 \times \mathbf{T}_{257} + \mathbf{T}_{256}. \tag{55}$$

Step 4 Obtain the permutation vector $\boldsymbol{p} = \{p_i\}_{i=1}^{256\times256}$ by stretching the matrix $\mathbf{T}_p$ row by row.

$$\begin{cases} ind = (i-1) \times 256 + j \\ p_{ind} = T_p(i,j) \end{cases}, 1 \le i \le 256, 1 \le j \le 256, \tag{56}$$

where ind is the index of the permutation vector $\boldsymbol{p}$.

2) If the size of a plain gray image is 512 × 512, the location vector $\boldsymbol{p} = \{p_i\}_{i=1}^{512\times512} = [0, 1, 2, \cdots, 512 \times 512 - 1]$ can be expanded to $n = 3$ digits and three chosen plain gray images are needed for the attack.

$$\mathbf{A}_2 = \begin{bmatrix} (0)(0)(0) & \cdots & (0)(0)(255) & (0)(1)(0) & \cdots & (0)(1)(255) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ (0)(254)(0) & \cdots & (0)(254)(255) & (0)(255)(0) & \cdots & (0)(255)(255) \\ (1)(0)(0) & \cdots & (1)(0)(255) & (1)(1)(0) & \cdots & (1)(1)(255) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ (3)(254)(0) & \cdots & (3)(254)(255) & (3)(255)(0) & \cdots & (3)(255)(255) \end{bmatrix}_{512\times512}. \tag{57}$$

Then, three matrices with entries $0, 1, 2, \cdots, 255$ are obtained by splitting matrix $\mathbf{A}_2$ into three bit-plane images. If the two chosen plain images are assumed to be $\mathbf{I}_{256} = \{I_{256}(i,j)\}_{i=1,j=1}^{M,N}$, $\mathbf{I}_{257} = \{I_{257}(i,j)\}_{i=1,j=1}^{M,N}$ and $\mathbf{I}_{258} = \{I_{258}(i,j)\}_{i=1,j=1}^{M,N}$, bit-plane 0 is assigned to $\mathbf{I}_{256}$; bit-plane 1 is assigned to $\mathbf{I}_{257}$; and bit-plane 2 is assigned to $\mathbf{I}_{258}$, respectively. The permutation equivalent mapping is calculated in the same manner shown in 1) above.

To illustrate this process, an image of 256 × 256 is used as an example, and we still want to attack its cipher image. From Fig. 10, the permutation vector $\boldsymbol{p}$ can be acquired from the chosen plain image $\mathbf{I}_{255}$ of size 256 × 256 and $\mathbf{I}_{256}$ of size 256 × 256. Fig. 11 shows the results for obtaining permutation equivalent mapping. The images from the first column to the last are the plain images, cipher images, the images after the elimination of the two-level diffusion effect, the images after reversing the shift rows transformation and the retrieved permutation-only images, respectively. The first row shows the conversion process for $\mathbf{I}_{256}$ and the second row shows the conversion process for $\mathbf{I}_{255}$. Taking the plain images (a1), (b1) and (c1) in Fig. 11 as the input images, the ciphertext images (a2), (b2) and (c2) in Fig. 11 can be obtained through the original encryption system (see Section 2). Then, the images after eliminating the diffusion effect, (a3), (b3) and (c3) in Fig. 11 are obtained by eliminating the two-level diffusion effect (see Section 3.1). After that, the permutation-only images (a4), (b4) and (c4) in Fig. 11 can be obtained by reverse shift rows transformation (see Section 3.2) and boundary pixels substitution (see Section 3.3).

From Eq. (47), we can see each column of $\mathbf{I}_{256}$ has the same elements. After the permutation of HCST, its columns get permutated first and then the rows become permutated. As a result, in Fig. 11(a5), the corresponding retrieved permutation-only image $\mathbf{I}_{256}$ is full of diagonal stripes. In contrast, elements in each row of $\mathbf{I}_{257}$ are the same. After the permutation of HCST, its columns are not messed up, but its rows are messed up. Therefore, the corresponding retrieved permutation-only image $\mathbf{T}_{257}$ in Fig. 11(b5) is also full of diagonal stripes. Additionally, it can be still seen

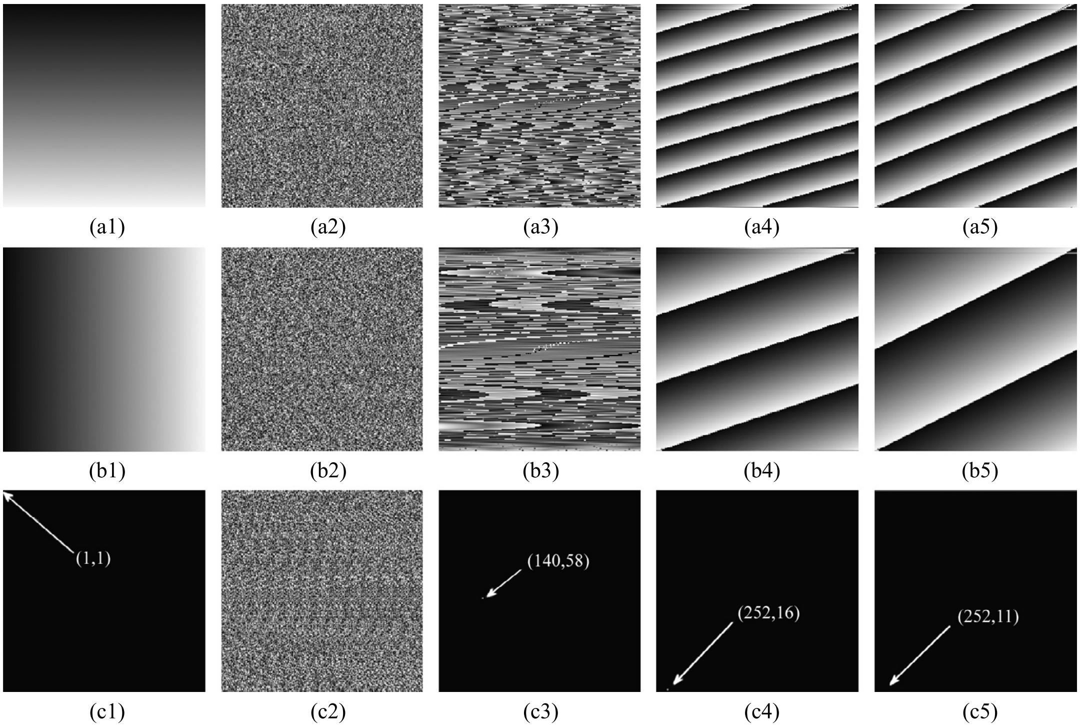


Fig. 11. CPA for obtaining the permutation equivalent mapping: images from the first column to the last are the plain images, cipher images, the images after the elimination of the two-level diffusion effect, the images after reversing the shift rows transformation and the retrieved permutation-only images, respectively.

that the left pixel of the diagonal stripe has a small value and a darker color. In Fig. 11(c1), only the first element of the plain image is nonzero. After the elimination of the two-level diffusion effect, the image in Fig. 11(c3) has only one nonzero pixel $(140, 58)$. Then after the inverse diagonal scanning transformation, only the element $(252, 16)$ is a nonzero pixel. From Section 3.3, it is known that the pixels in row 252 should move 251 bytes to the right circle. Therefore, it is shown in Fig. 11(c5) that the retrieved permutation-only image includes only one nonzero pixel located at $(252, 11)$. This location is the result of moving 5 bytes to the left or recycling mobile 251 bytes to the right. Based on Eq. (46), elements of the permutation matrix $\mathbf{T}_p$ can be obtained by combining Fig. 11(a5) and Fig. 11(b5). The obtained matrix $\mathbf{T}_p$ falls within scope $[0, 1, 2, \cdots, 256 \times 256 - 1]$ without repetition. These results test the correctness of our cryptanalysis method by measuring the permutation equivalent mapping.

3.5. *The overall deciphering process*

For any cipher image $\mathbf{R} \in [0, 255]$, a flowchart of the attack strategy is shown in Fig. 2, and the attack steps are described as follows:

Step 1 Obtain the matrix **Sd** via a chosen plain image (Attack 1 in Fig. 2) as stated in Section 3.1.1. Based on Proposition 1 and Proposition 2, the matrix **Sd** is exactly the cipher image of the plain image full of zero pixels.

Step 2 Obtain the map of diagonal scanning transformation **X** and **Y** by simulating the process of transformation in Fig. 6.

Step 3 Eliminate the two-level diffusion effect through the known matrix **Sd**, **X** and **Y** using the following elimination sub-steps.

Sub-Step 1 Calculate the cipher image $\mathbf{R}'$ by **R** and **Sd** with the XOR operation.

Sub-Step 2 Obtain the cipher image $\mathbf{O}'$ by the XOR operation for the row of $\mathbf{R}'$.

Sub-Step 3 Obtain the cipher image $\mathbf{D}'$ through the inverse diagonal scanning transformation on $\mathbf{O}'$ with the known maps **X** and **Y**, as described in Section 3.1.4.

Sub-Step 4 Obtain the cipher image $\mathbf{L}'$, which is the result of eliminating the two-level diffusion effect.

Step 4 Obtain the cipher image $\mathbf{P}'$ by the shift rows transformation on $\mathbf{L}'$, as described in Section 3.2.

Step 5 Calculate the boundary replacement matrix **Bp** via 256 chosen plain images (Attacks 1-256 in Fig. 2), as discussed in Section 3.3.

Step 6 Calculate the permutation-only image $\mathbf{T}'$ from the calculated boundary replacement matrix **Bp** and the cipher image $\mathbf{P}'$.

Step 7 Obtain the permutation equivalent mapping with known variables in Section 3.4.

Step 8 Obtain the recovered plain image from the calculated $\boldsymbol{p}$ and $\mathbf{P}'$ using the following inverse permutation sub-steps.

Sub-Step 1 Stretch the permutation-only image $\mathbf{P}' \in [0, 255]$ into vectors $\boldsymbol{V} = \{V(i)\}_{i=1}^{MN}$ row by row.

Sub-Step 2 Descramble the vector $\boldsymbol{V}$ above to generate vectors $\boldsymbol{V}' = \{V'(i)\}_{i=1}^{MN}$ using Eq. (49).

$$V'\big(p(i) + 1\big) = V'(i) \quad 1 \le i \le MN. \tag{58}$$

Sub-Step 3 Rearrange the new vectors $\boldsymbol{V}'$ above into the recovered plain image $\mathbf{I}'$ row by row.

The original encryption scheme is performed based on the pixel level, which can be applied to any type of image of any size if it has a gray format. Therefore, any image database that contains gray images is available. The USC-SIPI image database [41], which is widely used in image processing and used to verify the validity of the cryptanalysis work. The simulation results show that our proposed cryptanalysis can decrypt all the gray image cryptographic images. To illustrate our simulation results, three cipher images of 256 × 256 are used for verification. Three lines in Fig. 12 show experimental images entitled “Peppers”, “Chemical plant”

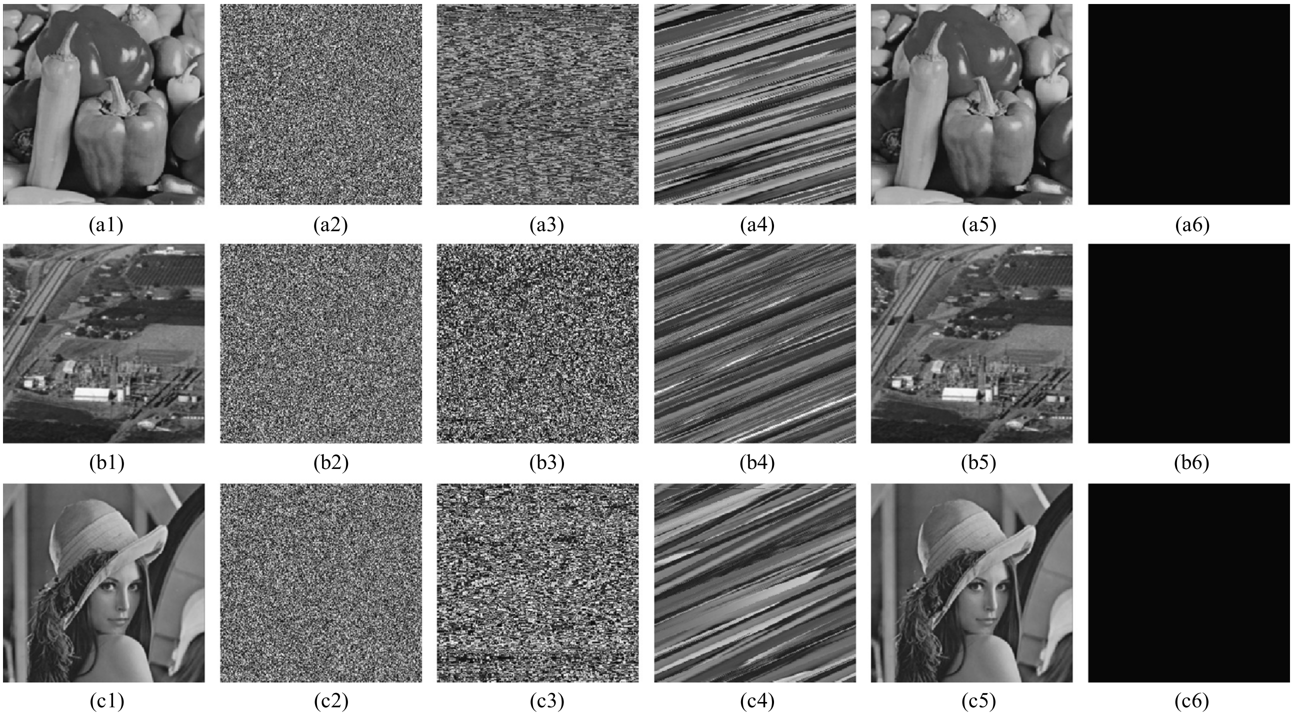


Fig. 12. The validity of the proposed cryptanalysis: images from column 1 to column 6 show plain images, cipher images, images with the diffusion effect eliminated, retrieved permutation-only images, final restored images and the XOR results for column 1 and column 5.

and “Lena”, respectively. From column 1 to column 5, plain images, cipher images, images with the diffusion effect eliminated, retrieved permutation-only images and restored images are displayed, respectively. Taking the plain images (a1), (b1) and (c1) in Fig. 12 as the input images, the ciphertext images (a2), (b2) and (c2) in Fig. 12 can be obtained through the original encryption system (see Section 2). Then, images (a3), (b3) and (c3) in Fig. 12 are obtained by eliminating the two-level diffusion effect (see Section 3.1). After that, the permutation-only images (a4), (b4) and (c4) in Fig. 12 can be obtained by reverse shift rows transformation (see Section 3.2) and boundary pixels substitution (see Section 3.3). Finally, the restored images (a4), (b4) and (c4) in Fig. 12 can be obtained through permutation equivalent mapping (see Section 3.4). Fig. 12 shows that the restored images in column 5 and the plain images in column 1 look completely identical. In fact, the images are shown in column 6 after applying the XOR operations between the plain images and corresponding recovered images, each of which has all zero pixels. This result demonstrates that there is not one pixel that is different between the plain image and corresponding restored image, which shows that all pixels in the original image have been restored.

3.6. Computational complexity analysis

For our cryptanalysis work, $256 + \lceil n \rceil$ plain images are needed for an attack in which one chosen plaintext image full of 0 pixels is used to obtain the matrix **Sd**, 255 chosen plaintext images are applied to compute the equivalent substitution matrix **Bp**, and $\lceil n \rceil$ chosen plaintext images are used to obtain the permutation map. For a cipher image of size 256×256, the number of chosen plaintext images is 258, and the time complexity is $O(258 \times 256 \times 256)$ close to $O(2^{24})$. Therefore, the number of at tacks is mainly used to solve the replacement matrix.

All the experiments were executed on a personal computer equipped with an Intel® Core™ i7-7500U 2.90 GHz CPU and 8 GB memory capacity. We use MATLAB R2016b to do all simulation experiments (the source code is available at https://github.com/ZhouKanglei/Cryptanalysis-of-MHM). The method we proposed for the execution time for different types of images is shown in Table 1. To ensure the accuracy of the experiment, different images were simulated many times, and the average value was obtained. From Table 1, we can conclude that the deciphering period and encryption period increase as the size of the image increases over time. Obviously, the equivalent replacement cycle for solving the replacement matrix is the main factor that restricts the deciphering efficiency.

Table 1
Execution time (seconds).

Image size	Encryption	Deciphering	
		Replacement cycle	Others
256×256	0.3091	2.1990	0.2007
512×512	1.2683	10.5439	0.9685
1024×1024	4.5026	49.5824	4.7596

From these results, in terms of time complexity analysis and actual running time analysis, solving the equivalent replacement matrix **Bp** costs a lot in the deciphering process. So, if the boundary substitution is not considered, the number of chosen plaintext images is $1 + \lceil n \rceil$. Meanwhile, the actual deciphering period is greatly reduced from Table 1. Additionally, take the three images used in Section 3.5 as an example in which three lines in Fig. 13 show experimental images entitled “Peppers”, “Chemical plant” and “Lena”, respectively. From column 1 to column 5, plain images, cipher images, images with the diffusion effect eliminated, retrieved permutation-only images and restored images are displayed, respectively. Taking the plain images (a1), (b1) and (c1) in Fig. 13 as the input images, the ciphertext images (a2), (b2) and (c2) in Fig. 13 can be obtained through the original encryption system (see Section 2). Then, images (a3), (b3) and (c3) in Fig. 13 are obtained by eliminating the two-level diffusion effect (see Section 3.1). After that, the permutation-only images (a4), (b4) and (c4) in Fig. 13 can be obtained by reverse shift rows transformation (see Section 3.2). Finally, the restored images (a4), (b4) and (c4) in

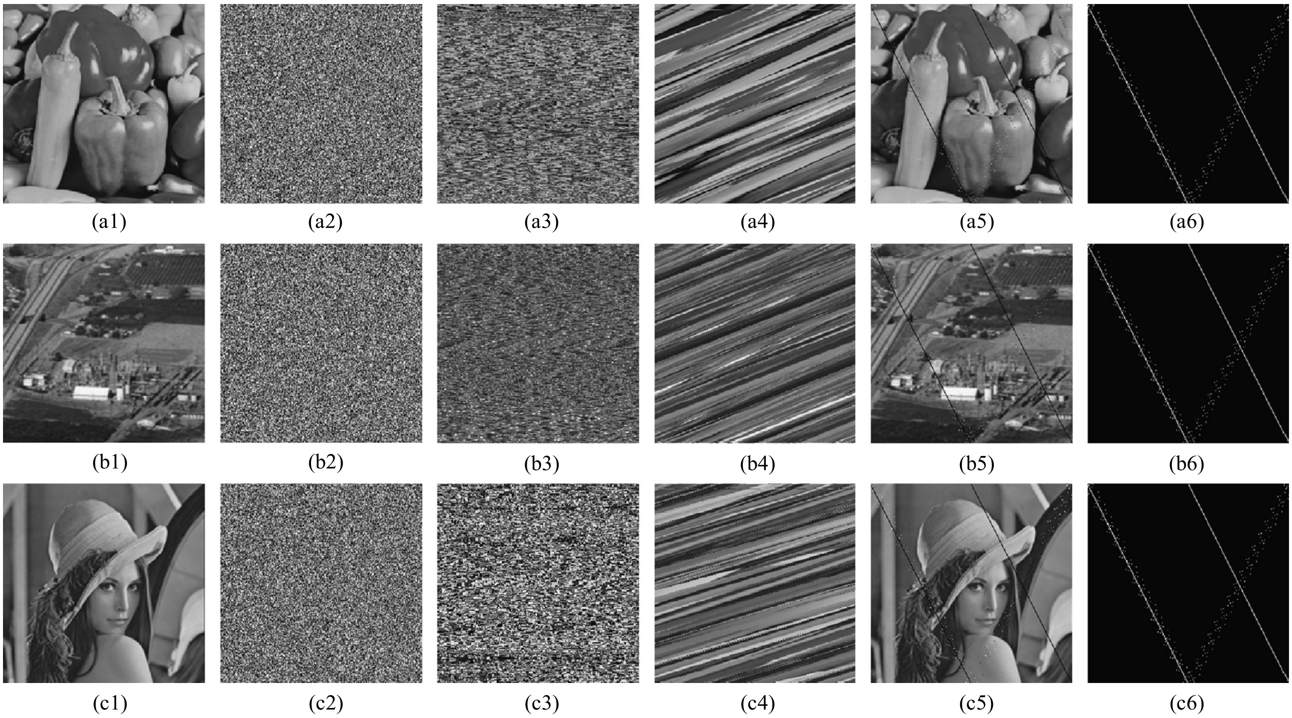


Fig. 13. The validity of the proposed cryptanalysis without the replacement: images from column 1 to column 6 show plain images, cipher images, images with the diffusion effect eliminated, retrieved permutation-only images, final restored images and the XOR results for column 1 and column 5, respectively.

Fig. 13 can be obtained through permutation equivalent mapping (see Section 3.4). Fig. 13 shows that the restored images in column 5 and the plain images in column 1 do not look very different. Furthermore, the images are shown in column 6 after applying the XOR operations between the plain images and corresponding recovered images, each of which have the 766 nonzero pixels. Nevertheless, the restored pixels are close to 98.83%. Therefore, the attack can be considered a success. Meanwhile, the deciphering efficiency is greatly improved, which is suitable for large-scale deciphering.

3.7. Method comparison

The confusion and diffusion structure used in the original encryption scheme has been cryptanalyzed by multiple methods. Analysis of the experiment shows that the computational complexity to break the permutation phase is $O(\lceil MN \log_c(MN) \rceil)$ [39]. Additionally, some researchers discussed in [32,42,43] that one round or several rounds of confusion and diffusion schemes can be broken using CPA. While these cryptanalysis approaches have simple calculations, they can break plaintext-related encryption schemes [32]. It is worth noting that we only use one ciphertext image of the known plaintext image to eliminate diffusion effects. Given the boundary pixels substitution, we need 255 chosen plaintext images, which will add to the complexity of our work. Therefore, we can choose to ignore the substitution and still obtain good results.

4. Conclusion

This paper attacked a chaotic image encryption scheme proposed recently, which is based on a modified Henon map using hybrid chaotic shift transform. It was claimed that the original encryption scheme could resist several known attacks. However, by using our proposed method, the original encryption scheme can be effectively cracked with $256 + \lceil n \rceil$ chosen plaintext images. Furthermore, only $1 + \lceil n \rceil$ chosen plaintext images are required if the boundary pixels are not considered. Experiment verifies the effectiveness of our cryptanalysis. The execution time is also satisfactory.

In order to improve the security of the cryptosystem, we give the following suggestions: 1) In the HCST and boundary pixels substitution phases, the initial values and parameters should not be the same. 2) It can be proved that the two-level diffusion in the original scheme is equivalent to one-level diffusion, which reduces the security of the cryptosystem greatly. Therefore, it is necessary to improve the structure of the encryption. 3) The encryption key should be associated to the plaintext images so that the attacker cannot reveal consistent key streams from different chosen plaintext images.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported through the National Natural Science Foundation of China (Grant Nos. 61602158, U1604156), the Science Foundation for the Excellent Youth Scholars of Henan Normal University (Grant No. YQ201607), the Science and Technology Research Project of Henan Province (Grant No. 182102210374), and the PhD Scientific Research Foundation of Henan Normal University (Grant No. 5101119170143).

References

[1] Y. Zhang, D. Xiao, W. Wen, H. Nan, Cryptanalysis of image scrambling based on chaotic sequences and Vigenère cipher, Nonlinear Dyn. 78 (1) (2014) 235–240.
[2] F. Özkaynak, S. Yavuz, Analysis and improvement of a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system, Nonlinear Dyn. 78 (2) (2014) 1311–1320.
[3] Y.G. Yang, P. Xu, R. Yang, et al., Quantum hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption, Sci. Rep. 6 (2016) 19788.

[4] Z. Hua, Y. Zhou, Design of image cipher using block-based scrambling and image filtering, Inf. Sci. 396 (2017) 97–113.
[5] X.Y. Wang, L. Yang, R. Liu, A. Kadir, A chaotic image encryption algorithm based on perceptron model, Nonlinear Dyn. 62 (2010) 615–621.
[6] Y.S. Zhang, D. Xiao, Y.L. Shu, J. Li, A novel image encryption scheme based on a linear hyperbolic chaotic system of partial differential equations, Signal Process. Image Commun. 28 (2013) 292–300.
[7] Z. Hua, Y. Zhou, Image encryption using 2D logistic-adjusted-sine map, Inf. Sci. 339 (2016) 237–253.
[8] M. Li, D. Xiao, Y.S. Zhang, H. Liu, Attack and improvement of the joint fingerprinting and decryption method for vector quantization images, Signal Process. 99 (2014) 17–28.
[9] H. Liu, Y. Liu, Security assessment on block-Cat-map based permutation applied to image encryption scheme, Opt. Laser Technol. 56 (1) (2014) 313–316.
[10] H. Fan, M. Li, D. Liu, K. An, Cryptanalysis of a plaintext-related chaotic RGB image encryption scheme using total plain image characteristics, Multimed. Tools Appl. 77 (15) (2018) 20103–20127.
[11] I.S. Sam, A novel image cipher based on mixed transformed logistic maps, Multimed. Tools Appl. 56 (2) (2012) 315–330.
[12] E.Y. Xie, C. Li, S. Yu, J. Lu, On the cryptanalysis of Fridrich's chaotic image encryption scheme, Signal Process. 132 (2017) 150–154.
[13] Y. Zhang, D. Xiao, W. Wen, M. Li, Cryptanalyzing a novel image cipher based on mixed transformed logistic maps, Multimed. Tools Appl. 73 (3) (2014) 1885–1896.
[14] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, Int. J. Bifurc. Chaos 8 (06) (1998) 1259–1284.
[15] W. Zhang, K.W. Wong, H. Yu, Z.L. Zhu, A symmetric color image encryption algorithm using the intrinsic features of bit distributions, Commun. Nonlinear Sci. Numer. Simul. 18 (3) (2013) 584–600.
[16] Z.L. Zhu, W. Zhang, K.W. Wong, H. Yu, A chaos-based symmetric image encryption scheme using a bit-level permutation, Inf. Sci. 181 (6) (2011) 1171–1186.
[17] H. Zhu, C. Zhao, X. Zhang, L. Yang, An image encryption scheme using generalized Arnold map and affine cipher, Optik, Int. J. Light Electron Opt. 125 (22) (2014) 6672–6677.
[18] E. Vaferi, R. Sabbaghi-Nadooshan, A new encryption algorithm for color images based on total chaotic shuffling scheme, Optik, Int. J. Light Electron Opt. 126 (20) (2015) 2474–2480.
[19] S.M. Wadi, N. Zainal, Decomposition by binary codes-based speedy image encryption algorithm for multiple applications, IET Image Process. 9 (5) (2015) 413–423.
[20] X. Wang, H.L. Zhang, A color image encryption with heterogeneous bit-permutation and correlated chaos, Opt. Commun. 342 (2015) 51–60.
[21] Y.Q. Zhang, X.Y. Wang, A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice, Inf. Sci. 273 (8) (2014) 329–351.
[22] Y. Zhou, W. Cao, C.L. Philp Chen, Image encryption using binary bitplane, Signal Process. 100 (2014) 197–207.
[23] Y. Liu, L.Y. Zhang, J. Wang, Y. Zhang, K. Wong, Chosen-plaintext attack of an image encryption scheme based on modified permutation–diffusion structure, Nonlinear Dyn. 84 (4) (2016) 2241–2250.
[24] K.W. Wong, S.H. Kwok, W.S. Law, A fast image encryption scheme based on chaotic standard map, Phys. Lett. A 372 (15) (2008) 2645–2652.
[25] A.V. Diaconu, Circular inter–intra pixels bit-level permutation and chaos-based image encryption, Inf. Sci. 355 (2016) 314–327.
[26] H. Fan, M. Li, Cryptanalysis and improvement of chaos-based image encryption scheme with circular inter-intra-pixels bit-level permutation, Math. Probl. Eng. 2017 (2017) 1–11.
[27] C. Li, Y. Zhang, R. Ou, et al., Breaking a novel colour image encryption algorithm based on chaos, Nonlinear Dyn. 70 (4) (2012) 2383–2388.
[28] X. Wang, L. Teng, X. Qin, A novel colour image encryption algorithm based on chaos, Signal Process. 92 (4) (2012) 1101–1108.
[29] C. Li, B. Feng, J. Lu, Cryptanalysis of a Chaotic Image Encryption Algorithm Based on Information Entropy, 2018.
[30] G. Ye, C. Pan, X. Huang, et al., A chaotic image encryption algorithm based on information entropy, Int. J. Bifurc. Chaos 28 (01) (2018) 9.
[31] H. Fan, M. Li, D. Liu, K. An, Cryptanalysis of a plaintext-related chaotic RGB image encryption scheme using total plain image characteristics, Multimed. Tools Appl. 77 (15) (2017) 20103–20127.
[32] M.A. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez, et al., A RGB image encryption algorithm based on total plain image characteristics and chaos, Signal Process. 109 (2015) 119–131.
[33] Y. Zhang, D. Xiao, W. Wen, et al., Breaking an image encryption algorithm based on hyper-chaotic system with only one round diffusion process, Nonlinear Dyn. 76 (3) (2014) 1645–1650.
[34] B. Norouzi, S. Mirzakuchaki, S.M. Seyedzadeh, et al., A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process, Multimed. Tools Appl. 71 (3) (2014) 1469–1497.
[35] C.Y. Song, Y.L. Qiao, X.Z. Zhang, An image encryption scheme based on new spatiotemporal chaos, Optik, Int. J. Light Electron Opt. 124 (2013) 3329–3334.
[36] H.I. Hsiao, J. Lee, Color image encryption using chaotic nonlinear adaptive filter, Signal Process. 117 (C) (2015) 281–309.
[37] L. Zhen, C. Peng, L. Li, et al., A novel plaintext-related image encryption scheme using hyper-chaotic system, Nonlinear Dyn. 94 (2) (2018) 1319–1333.
[38] S.J. Sheela, K.V. Suresh, D. Tandur, Image encryption based on modified Henon map using hybrid chaotic shift transform, Multimed. Tools Appl. 77 (19) (2018) 25223–25251.
[39] A. Jolfaei, X. Wu, V. Muthukkumarasamy, On the security of permutation-only image encryption schemes, IEEE Trans. Inf. Forensics Secur. 11 (2) (2016) 235–246.
[40] G. Hanchinamani, L. Kulkarni, An efficient image encryption scheme based on a Peter De Jong chaotic map and a RC4 stream cipher, 3D Res. 6 (3) (2015) 1–15.
[41] USC-SIPI Image Database, University of South California, Signal and Image Processing Institute, http://sipi.usc.edu/database. (Accessed 28 August 2017).
[42] E. Solak, C. Çokal, O.T. Yildiz, T. Biyikoglu, Cryptanalysis of Fridrich's chaotic image encryption, Int. J. Bifurc. Chaos 20 (5) (2010) 1405–1413.
[43] L.Y. Zhang, Y. Liu, F. Pareschi, et al., On the security of a class of diffusion mechanisms for image encryption, IEEE Trans. Cybern. 48 (4) (2018) 1163–1175.

Kanglei Zhou, a junior undergraduate, is currently studying at College of Computer and Information Engineering, Henan Normal University, China. His research interest is mainly in information security.

Minghui Xu was admitted to the College of Computer and Information Engineering of Henan Normal University in 2016 to pursue the bachelor's degree in engineering. His research interests include multimedia security, deep learning, compressive sensing.

Jidong Luo was admitted to the School of Computer and Information Engineering of Henan Normal University in 2016 to pursue the bachelor's degree in engineering. His research interests include information hiding, deep learning, compressive sensing.

Haiju Fan received the master's degree in science from the College of Electronic Information Engineering, Beihang University, Beijing, China, in 2005, and the Ph.D. degree from Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2017. She is currently an Associate Professor with the College of Computer and Information Engineering, Henan Normal University. Her research interests include multimedia security, information hiding, and compressive sensing.

Ming Li received the master's degree in science from the College of Physics and Information Engineering, Henan Normal University, Henan, China, in 2010, and the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2014. He is currently an Associate Professor with the College of Computer and Information Engineering, Henan Normal University. His research interests include multimedia security, information hiding, and compressive sensing.