Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-14921-0

Security analysis of a color image encryption based on bit-level and chaotic map

Heping Wen[1,2,3] · Ruiting Chen[1] · Jieyi Yang[1] · Tianle Zheng[1] · Jiahao Wu[1] · Wenxing Lin[1] · Huilin Jian[1] · Yiting Lin[1] · Linchao Ma[1] · Zhen Liu[1] · Chongfu Zhang[1,2]

Received: 2 November 2021 / Revised: 9 January 2023 / Accepted: 22 February 2023

Abstract

In recent years, various chaotic image encryption algorithms have been proposed, but their security issues still need to be studied and reviewed. This paper implements a cryptographic security analysis on a color image encryption algorithm based on bit-level and improved one-dimensional chaotic map (CIEA-IOCM). In CIEA-IOCM, using the pseudo-random sequences generated by an improved one-dimensional chaotic map, bit-level permutation-diffusion and linear transformation are successively performed to get its cipher image from a plain image. Based on some seemingly good performance analysis results, CIEA-IOCM claimed that it can withstand various common attacks. However, after a cryptanalysis, we found that it can be completely cracked by a chosen-plaintext attack method. The most essential reason is that there are equivalent keys for both bit-level permutation and diffusion processes, and its linear transformation can be combined and simplified. Thus, we achieved a successful security attack by obtaining the equivalent keys. Moreover, the experimental results verify the effectiveness and efficiency of our method. Therefore, our research work can provide some positive and valuable references for improving the security of chaotic image encryption algorithms.

Keywords Chaos · Cryptanalysis · Image encryption · Chosen-plaintext attack

✉ Heping Wen
wenheping@uestc.edu.cn

Ruiting Chen
ruitingchen@stu.zsc.edu.cn

Yiting Lin
Dr.YitingLin@gmail.com

[1] University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan 528402, China

[2] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

[3] Guangdong Provincial Key Laboratory of Information Security Technology of Sun Yat-sen University, Guangzhou 510006, China

Published online: 23 May 2023

1 Introduction

The continuous advancement of network communication technology has greatly improved people's lifestyle and the more of production. When people are enjoying the convenience brought by technological development, sensitive information is easily leaked. Once the information leaked, it may threat people's life and property. Therefore, information leakage is a matter of great concern. By discovering that image information is more intuitive and interactive, people can more intuitively obtain the information they want to know, and for this reason people prefer to use images to record objective things in daily life. Attackers usually use network vulnerabilities and other flaws to steal sensitive information. Thus, image encryption technology is becoming more and more important for image owners in various fields, and its security issues have received extensive attention [22, 33]. To this end, researchers have proposed a number of new image encryption schemes, such as chaos theory [3, 10, 25], DNA coding [5, 9, 25, 32], cellular automata [6, 23, 26], quantum theory [7, 24, 29], and so on [1, 12, 13]. In 1963, American meteorologist Lorenz published a classic paper on chaos theory, and found a very complicated phenomenon in a certain nonlinear system which is called a chaotic phenomenon. The characteristics of chaotic systems include randomness, topological transitivity, and density of periodic points, which are consistent with the basic requirements of traditional cryptographic algorithms [16, 19]. As a result, the image encryption technology based on chaos theory has become a research hotspot in recent years [10, 18, 28, 29, 39]. However, due to the lack of effective cryptanalysis of many encryption algorithms, some of them have been deciphered for their security flaws [33, 35]. In summary, it is necessary to analyze the security of existing encryption algorithms [27, 30, 31, 36].

In recent years, the bit-level chaotic image encryption algorithms [2, 4, 21] have attracted the attention of researchers, and many related encryption algorithms have been proposed one after another [18, 39]. In 2011, Ref. [39] proposed an image encryption algorithm based on bit-level permutation. During the permutation process, not only the position but also the value of the pixel are changed which has the effect of confusion and diffusion. Inspired by this, in the same year, Ref. [15] proposed a diffusion encryption algorithm based on bit-level permutation and high-dimensional chaos, which has a larger key space. However, in 2014, Ref. [35] pointed out that the algorithm in Ref. [39] is insecure and has the defect of being vulnerable to plaintext attacks. In order to solve this problem, in 2014, Ref. [37] proposed a bit-level image encryption algorithm associated with plaintext to enhance the security of the algorithm. Since then, image encryption technology based on bit-level processing has received more attention [11, 18]. In 2015, Ref. [8] proposed an image encryption algorithm combining global and local bit-level permutation. In 2016, Ref. [34] proposed an image encryption algorithm with the algorithm structure of "permutation-diffusion-permutation" based on bit-level image permutation encryption. In 2017, Ref. [14] proposed an image encryption algorithm combining pixel-level and bit-level permutation. In 2018, Ref. [20] further proposed an image encryption algorithm that combines DNA encoding, pixel-level and bit-level permutation. With the improvement of the levels of analysis and the design of chaotic cipher algorithm, the difficulty of algorithm deciphering has increased. However, there are still security flaws in some algorithms that cannot withstand common password attacks. Therefore, the study of chaotic cryptographic algorithms used to encrypt and protect image files is of importance and value theory and practical application.

In 2019, a bit-level color image encryption algorithm based on the improved one-dimensional chaotic map was proposed [17]. In the color image encryption algorithm, using an improved one-dimensional chaotic map system (CIEA-IOCM), using the pseudo-random sequences generated by the improved one-dimensional chaotic map system, bit-level permutation, bit-level pixel diffusion and linear transformation are successively performed to

get cipher images from the plain images. Meanwhile, the relevant pixel values, histograms and other experimental analyses are given to verify its security performance. However, from the perspective of cryptanalysis, we found some security defects as follows:

- There are some equivalent keys. CIEA-IOCM encrypts the image using the pseudo-random sequence generated by the improved one-dimensional chaotic map system. However, these sequences are irrelevant to the plaintext. Therefore, the sequences can be seen as the equivalent keys.
- The algorithm of the diffusion part and linear transformation is unsafe. The diffusion part is to perform the XOR operation and the linear transformation part is a round of linear transformation to the right. It is noted that their process can be completely based on the commutative law of algebraic properties, and the two parts can be combined into a new diffusion matrix. Since then, we can easily crack the new diffusion matrix by choosing an image whose pixel values are zero and get its corresponding cipher images.

Based on the above two points, CIEA-IOCM cannot resist against a chosen-plaintext attack method with the divide-and-conquer strategy. More specifically, in the case of the chosen-plaintext attack, firstly an equivalent diffusion key is obtained, and then an equivalent permutation key is obtained. Finally, the original images can be restored from the encrypted images with the equivalent keys.

2 The encryption algorithm under study

This section will introduce the improved one-dimensional chaotic map system used in Ref. [17], and then the detailed steps of the CIEA-IOCM.

2.1 Improved one-dimensional chaotic map

Because of the simple structure, the one-dimensional chaotic maps are widely used in image encryption. Logistic map is one of the simple one-dimensional chaotic maps with complex chaotic behavior. It is represented by the following equation:

$$x_{n+1} = F_L(u, x_n) = u \times x_n \times (1 - x_n) \tag{1}$$

where u is a control parameter in the range of $(0, 4]$, x_0 is the initial value of chaotic map and x_n is the output chaotic sequence.

The improved one-dimensional chaotic map structure is derived from Ref. [17], given as:

$$\begin{aligned} x_n &= F(u, x_n, k) \\ &= mod((F_{chaos}(u, x_n) - F'_{chaos}(u, x_n)) \times G(k), 1) \end{aligned} \tag{2}$$

where the $G(k) = 2^k$, $9 \leq k \leq 16$, $F_{chaos}(u, x_n)$ is one of the existing one-dimensional chaotic maps mentioned above and $F(u, x_n, k)$ is a newly made chaotic map. $F'_{chaos}(u, x_n)$ is the function where u of the function $F_{chaos}(u, x_n)$ is replaced with $(4 - u)$. u is a control parameter in the range of $[0, 2] \cup (2, 4]$.

Then, the improved logistic map equation is obtained by applying the structure of (2) as follow:

$$\begin{aligned} x_{n+1} =& mod((u \times x_n \times (1 - x_n) - (4 - u) \\ & \times x_n \times (1 - x_n)) \times 2^{12}, 1) \end{aligned} \tag{3}$$

where the parameter u is in the range of $[0, 2] \cup (2, 4]$ and x_0 is the initial value of the sequence.

2.2 Description of CIEA-IOCM

As shown in Fig. 1, the CIEA-IOCM includes five main parts: bit-level decomposition, bit-level permutation, bit-level pixel diffusion, linear transformation and bit-level combination. It is noted that bit-level decomposition adopts binary bit-level decomposition (BBD) method [38]. The main contents are briefly introduced as follows:

- **The key parameters:**
  The CIEA-IOCM uses (x_0, u, k, N_0, kd, rp) six parameters as security keys. The N_0 is a number of iterations in (3), k participates in generating the diffusion matrix in (6), rp is the amount involved in rotating the matrix to the right in (8) and (x_0, u, k) are the initial values of the improved one-dimensional chaotic map system defined in (3).
- **Step 1.** Bit-level decomposition:
  The color image I with the size of $M \times N \times 3$ is divided into 3 image channels of R, G and B, and then 3 images are linked to make a grayscale image with the size of $M \times 3N$. If it is a grayscale image with the size of $M \times N$, it can be used without conversion. The grayscale image obtained above is converted into a one-dimensional image pixel matrix, and then it is converted into a one-dimensional image bit matrix $B = \{b_1,\ b_2,\ \cdots,\ b_{M\times 24N}\}$.
- **Step 2.** Bit-level permutation:
  The chaotic sequence X is generated in the CIEA-IOCM through the above improved logistic map in (3). It iterates $M \times 24N + N_0$ times, and discards the former N_0 elements to make a new sequence with $M \times 24N$ elements. Then, the permutation position matrix X' is obtained by sorting the chaotic sequence X in an ascending order:

$$X' = \{x'_1,\ x'_2,\ \cdots,\ x'_{M\times 24N}\} \tag{4}$$

  The permuted image bit matrix B' is obtained by using permutation position matrix X' and the image bit matrix B. The permuted image bit matrix B' can be expressed by the following equation:

$$B' = B(X'(i)) \tag{5}$$

  where $i = 1 \sim L$ and $L = M \times 24N$ represent the number of pixels in the three RGB image channels converted to the bit-level.
- **Step 3.** Bit-level pixel diffusion:

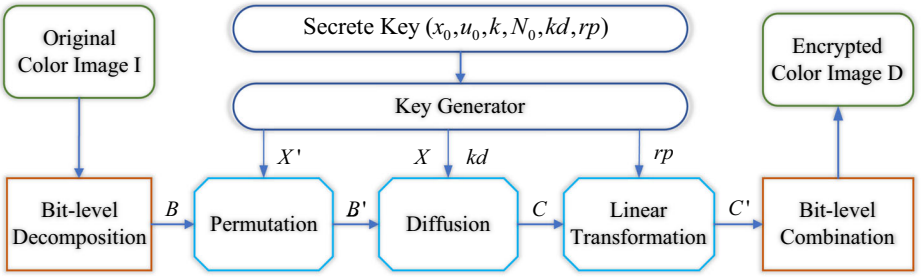


Fig. 1 The block diagram of CIEA-IOCM

By the following equation, the diffusion matrix $D' = \{d'_1, d'_2, \cdots, d'_{M\times 24N}\}$ is obtained, given as:

$$D'(i) = mod(floor(X(i) \times kd), 2) \tag{6}$$

where kd is a positive integer and the diffusion matrix D' consists of 0 and 1.

From the diffusion matrix D' and the permuted image bit matrix B', we obtain the encrypted image bit matrix $C = \{c_1, c_2, \cdots, c_{M\times 24N}\}$, by the following diffusion equation:

$$C(i) = bitxor(B'(i), D'(i)) \tag{7}$$

- **Step 4.** Linear transformation:

  We obtain a new encrypted image bit matrix $C' = \{c'_1, c'_2, \cdots, c'_{M\times 24N}\}$ through rotating the obtained matrix C to the right by the amount of units rp. The new image matrix bit C' is obtained as follows:

$$\begin{cases} C'(i+rp) = C(i) \\ i + rp \le M \times 24N \\ C'((i+rp) - M \times 24N) = C(i) \\ i + rp > M \times 24N \end{cases} \tag{8}$$

  where $rp \in [1, M \times 24N]$.
- **Step 5.** Bit-level combination:

  A two-dimensional encrypted image D is obtained by using the bit-level synthesis of the new encrypted image bit matrix C', and the size of the image D is $M \times N \times 3$.

3 Cryptanalysis and deciphering

3.1 Preliminary analysis of CIEA-IOCM

In this section, firstly, the security defects of the original algorithm are analyzed, and then a decoding method based on chosen-plaintext attack is proposed. Refering to the basic assumptions of cryptanalysis, all the contents of the cryptosystem are public, and the attackers only have the secret key unknown. Chosen-plaintext attack is a common but powerful cryptanalysis method. It is assumed that the attackers can arbitrarily choose the plaintext that is conducive to decryption and obtain the corresponding ciphertext. In the case of chosen-plaintext attack, the attacker can construct a special ordinary image (such as all black and all white images), and obtain the corresponding special image to analyze the target image.

From the perspective of mathematical derivation, it can combine the Step 3 of solving the diffusion matrix D' and the linear transformation of the Step 4 in the Section 2.2 to obtain a new diffusion matrix D''. The reason is that the algorithm only performs one round of linear transformation and only generates the diffusion matrix XOR operation, which satisfies the commutative law of algebraic properties. The new diffusion matrix D'' is defined as:

$$\begin{cases} D''(i+rp) = mod(floor(X(i) \times kd), 2) \\ \qquad\qquad\qquad\qquad\qquad i + rp \le M \times 24N \\ D''((i+rp) - M \times 24N) = mod\ (floor(X(i) \times kd), 2) \\ \qquad\qquad\qquad\qquad\qquad i + rp > M \times 24N \end{cases} \tag{9}$$

Subsequently, we perform the diffusion operation to get directly bit matrix C', given as:

$$C'(i) = bitxor(B'(i),\ D''(i)) \tag{10}$$

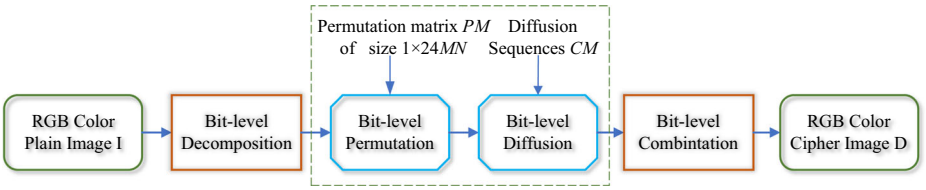


Fig. 2 The block diagram of an equivalent simplified CIEA-IOCM

Finally, perform bit-level synthesis on the obtained encrypted image bit matrix C' to obtain a two-dimensional encrypted image. At this time, the algorithm has completed the encryption.

The algorithm structure of CIEA-IOCM is actually a classic single-round permutation-diffusion. In addition, all the generation processes of chaotic-based pseudo-random sequences have nothing to do with ordinary images, which means that these sequences can be regarded as equivalent keys. The reason is that, in the case of a given secret key, these sequences are fixed for encrypting different plain images with the same size. Then, the CIEA-IOCM can be equivalently simplified as shown in Fig. 2, where PM is an equivalent permutation key and the diffusion sequences CM serve as an equivalent diffusion key.

In summary, under the scenario of chosen-plaintext attack and the strategy of divide-and-conquer, the equivalent keys can be obtained, and then the original image can be restored. Specifically, firstly, select some plain images with the same pixel values to eliminate the pernuation and get the corresponding cipher image, and then get the diffusion key. Subsequently, construct some special ordinary images with unequal element values and get the corresponding permuation images, and then obtain the permuation key through the mapping relationship. Finally, use the equivalent keys to restore the images.

3.2 Diffusion analysis

In this section, based on the chosen-plaintext attack, it is assumed that the plaintext image with the same pixel value is selected and the corresponding ciphertext image is obtained.

- **Step 1.** Choose the image I^0 with all-zero pixel values and obtain the corresponding cipher image C^0 to get the equivalent diffusion matrix CM.

  The reason for choosing the all-zero image is that no matter how complex the permuation process is, the matrix value is 0 after the image is permuted. Therefore, the permuation is ineffective, and in this way we can eliminate the diffusion to a relatively large extent. At this time, the (10) becomes:

$$C'(i) = D''(i) \tag{11}$$

  Then the equivalent diffusion matrix CM is equal to $C'(i)$ in (11)

3.3 Analysis of permutation part

Once the diffusion part is broken, CIEA-IOCM degenerates into a permutation-only cipher. According to the existing research, it cannot resist the chosen-plaintext attack. The basic idea is to construct a special ordinary image with unequal element values and obtain the corresponding permuted image. Taking 2×4 as an example, the process of solving PM is described below. Firstly, a chosen plain image and the corresponding permuted image are

given as:

$$I = \begin{bmatrix} 0 & 1 & 4 & 5 \\ 2 & 3 & 6 & 7 \end{bmatrix}; P = \begin{bmatrix} 5 & 4 & 1 & 7 \\ 3 & 0 & 2 & 6 \end{bmatrix}$$

Then, the permutation process can be described as follows:

$$\begin{bmatrix} 0 & 1 & 4 & 5 \\ 2 & 3 & 6 & 7 \end{bmatrix} \xrightarrow{PM} \begin{bmatrix} 5 & 4 & 1 & 7 \\ 3 & 0 & 2 & 6 \end{bmatrix}$$

where PM is the permutation matrix of size $M \times N$.

Finally, PM is determined as:

$$PM = \begin{bmatrix} (2,2) & (1,3) & (1,2) & (1,1) \\ (2,3) & (2,1) & (2,4) & (1,4) \end{bmatrix}$$

However, since the CIEA-IOCM is permuted in the bit-level, and the bit-level only has two values, 0 and 1. Therefore, PM has to be obtained in a different way. For an image with a size of $M \times N \times 3$, the number of pixels when they are converted into bit-level are $24MN$, so we can construct a virtual plaintext matrix vp with a size of $1 \times 24MN$, in which all pixel values are not equal and are arranged in ascending order, namely:

$$vp = [0,\ 1,\ 2,\ 3,\ \cdots,\ 24MN - 1] \tag{12}$$

Then, decompose the virtual plaintext matrix vp into N_C corresponding binary plaintext matrices according to the law of high and low bits, and get:

$$vp = \sum_{n=1}^{N_C} 2^{n-1} p_n = p_1 + 2p_2 + 2^3 p_3 + \cdots + 2^{N_C - 1} p_{N_c} \tag{13}$$

where $p_n = 0\ or\ 1$, $N_c = \lceil \log_2(1 \times 24MN) \rceil$, $\lceil . \rceil$ means rounding up operation and log means logarithm.

Secondly, according to the conditions of chosen-plaintext attack, an encryption machine is temporarily obtained, and the element position permuation encryption is employed on the N_C plaintext matrices p_n $(n = 1,\ 2,\ \cdots,\ N_c)$, and the corresponding N_C permuation matrices can be obtained as p'_n $(n = 1,\ 2,\ \cdots,\ N_c)$. Thirdly, according to (13), the virtual matrix vp' is composed from the N_C permuation matrices p'_n $(n = 1,\ 2,\ \cdots,\ N_c)$, given as:

$$vp' = \sum_{n=1}^{N_C} 2^{n-1} p'_n = p'_1 + 2p'_2 + 2^3 p'_3 + \cdots + 2^{N_C - 1} p'_{N_c} \tag{14}$$

Finally, by comparing the permutation correspondences of all elements in these two virtual matrices vp and vp', the equivalent permutation key PM can be obtained.

For example, the permutation process of the matrix I of size $2 \times 2 \times 3$ in the bit-level is visualized as shown in Fig. 3. According to the (13), $N_C = 7$ is obtained and vp is decomposed with the law of the high and low bits to obtain 7 corresponding binary matrices

$$I_{bit}^{1\times 96} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 \\ 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 \\ 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 \\ 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 & 81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 & 90 & 91 & 92 & 93 & 94 & 95 \end{bmatrix}$$

PM

$$P_{bit}^{1\times 96} = \begin{bmatrix} 11 & 1 & 71 & 8 & 68 & 65 & 62 & 59 & 49 & 52 & 93 & 43 & 44 & 17 & 27 & 94 & 55 & 53 & 15 & 50 & 57 & 60 & 63 & 66 \\ 69 & 9 & 72 & 32 & 2 & 29 & 12 & 24 & 90 & 46 & 5 & 86 & 19 & 81 & 75 & 35 & 36 & 37 & 38 & 76 & 39 & 77 & 82 & 20 \\ 40 & 78 & 83 & 87 & 21 & 6 & 47 & 91 & 41 & 25 & 13 & 30 & 3 & 79 & 33 & 73 & 84 & 88 & 22 & 10 & 0 & 70 & 7 & 67 \\ 64 & 61 & 58 & 48 & 51 & 92 & 42 & 16 & 26 & 54 & 14 & 56 & 31 & 95 & 28 & 4 & 18 & 80 & 34 & 74 & 85 & 45 & 89 & 23 \end{bmatrix}$$

Fig. 3 The visual flow chart of the position permuation of bit-level matrix elements with the size 1×96

as follows:

$$p_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$p_2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$p_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$p_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$p_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$p_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$p_7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Temporarily, by using the encryption machine shown in Fig. 3, the corresponding 7 binary permutation matrices are:

$$p'_1 = \begin{bmatrix} 1&1&1&0&0&1&0&1&1&0&1&1&0&1&1&0&1&1&1&0&1&0&1&0 \\ 1&1&0&0&0&1&0&0&0&0&1&0&1&1&1&1&0&1&0&0&1&1&0&0 \\ 0&0&1&1&1&0&1&1&1&1&1&0&1&1&1&1&0&0&0&0&0&0&1&1 \\ 0&1&0&0&1&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&1&1 \end{bmatrix}$$

$$p'_2 = \begin{bmatrix} 1&0&1&0&0&0&1&1&0&0&0&1&0&0&1&1&1&0&1&1&0&0&1&1 \\ 0&0&0&0&1&0&0&0&1&1&0&1&1&0&1&1&0&0&1&0&1&0&1&0 \\ 0&1&1&1&0&1&1&1&0&0&0&1&1&1&0&0&0&0&1&1&0&1&1&1 \\ 0&0&1&0&1&0&1&0&1&1&1&0&1&1&0&0&1&0&1&1&0&0&0&1 \end{bmatrix}$$

$$p'_3 = \begin{bmatrix} 0&0&1&0&1&0&1&0&0&1&1&0&1&0&1&0&0&1&1&1&0&0&1&0 \\ 1&0&0&0&0&1&1&0&0&1&1&1&0&0&0&0&1&1&1&1&1&1&0&1 \\ 0&1&0&1&1&1&1&0&0&0&1&1&0&1&0&0&1&0&1&0&0&1&1&0 \\ 0&1&0&0&0&1&0&0&0&1&1&0&1&1&1&1&0&0&0&0&1&1&0&1 \end{bmatrix}$$

$$p'_4 = \begin{bmatrix} 1&0&0&1&0&0&1&1&0&0&1&1&1&0&1&1&0&0&1&0&1&1&1&0 \\ 0&1&1&0&0&1&1&1&1&1&0&0&0&0&1&0&0&0&0&1&0&1&0&0 \\ 1&1&0&0&0&0&1&1&1&1&1&1&0&1&0&1&0&1&0&1&0&0&0&0 \\ 0&1&1&0&0&1&1&0&1&0&1&1&1&1&1&0&0&0&0&1&0&1&1&0 \end{bmatrix}$$

$$p'_5 = \begin{bmatrix} 0&0&0&0&0&0&1&1&1&1&1&0&0&1&1&1&1&1&0&1&1&1&1&0 \\ 0&0&0&0&0&1&0&1&1&0&0&1&1&1&0&0&0&0&0&0&0&0&1&1 \\ 0&0&1&1&1&0&0&1&0&1&0&1&0&0&0&0&1&1&1&0&0&0&0&0 \\ 0&1&1&1&1&1&0&1&1&1&0&1&1&1&1&0&1&1&0&0&1&0&1&1 \end{bmatrix}$$

$$p'_6 = \begin{bmatrix} 0&0&0&0&0&0&1&1&1&1&0&1&1&0&0&0&1&1&0&1&1&1&1&0 \\ 0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&1&1&1&0&1&0&0&0 \\ 1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&0&1&0&0&1&0&1&0&0&0&0&0&0&1&0&0&1&0&0 \end{bmatrix}$$

$$p'_7 = \begin{bmatrix} 0&0&1&0&1&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&1 \\ 1&0&1&0&0&0&0&0&1&0&0&1&0&1&1&0&0&0&0&1&0&1&1&0 \\ 0&1&1&1&0&0&0&1&0&0&0&0&0&0&1&0&1&1&1&0&0&0&1&0&1 \\ 1&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&1&1&0&1&0 \end{bmatrix}$$

Then according to the (14), the corresponding output virtual matrix vp' is obtained as:

$$vp' = \begin{bmatrix} 11&1&71&8&68&65&62&59&49&52&93&43&44&17&27&94&55&53&15&50&57&60&63&66 \\ 69&9&72&32&2&29&12&24&90&46&5&86&19&81&75&35&36&37&38&76&39&77&82&20 \\ 40&78&83&87&21&6&47&91&41&25&13&30&3&79&33&73&84&88&22&10&0&70&7&67 \\ 64&61&58&48&51&92&42&16&26&54&14&56&31&95&28&4&18&80&34&74&85&45&89&23 \end{bmatrix}$$

Finally, by comparing vp and vp', it is not difficult to obtain the equivalent permutation key PM. Therefore, the PM is determined as:

$$PM = \begin{bmatrix} (1,69)&(1,2)&(1,29)&(1,61)&(1,88)&(1,35)&(1,54)&(1,71)&(1,4)&(1,26)&(1,68)&(1,1) \\ (1,31)&(1,58)&(1,83)&(1,19)&(1,80)&(1,14)&(1,89)&(1,37)&(1,48)&(1,53)&(1,67)&(1,96) \\ (1,32)&(1,58)&(1,81)&(1,15)&(1,87)&(1,30)&(1,60)&(1,85)&(1,28)&(1,63)&(1,91)&(1,40) \\ (1,41)&(1,42)&(1,43)&(1,45)&(1,49)&(1,57)&(1,79)&(1,12)&(1,13)&(1,94)&(1,34)&(1,55) \\ (1,76)&(1,9)&(1,20)&(1,77)&(1,10)&(1,18)&(1,82)&(1,17)&(1,84)&(1,21)&(1,75)&(1,8) \\ (1,22)&(1,74)&(1,7)&(1,23)&(1,73)&(1,6)&(1,24)&(1,72)&(1,5)&(1,25)&(1,70)&(1,3) \\ (1,27)&(1,64)&(1,92)&(1,39)&(1,44)&(1,46)&(1,50)&(1,62)&(1,81)&(1,38)&(1,47)&(1,51) \\ (1,65)&(1,93)&(1,36)&(1,52)&(1,90)&(1,95)&(1,33)&(1,56)&(1,78)&(1,11)&(1,16)&(1,86) \end{bmatrix} \quad (15)$$

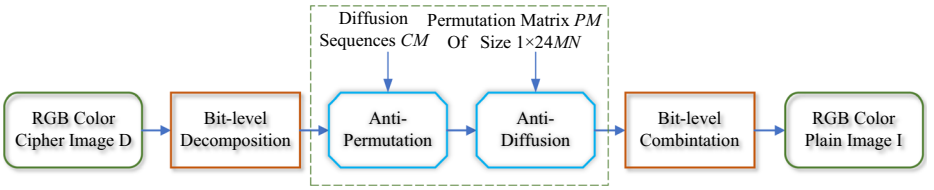


Fig. 4 The overall flow chart of attacking an improved chaotic system

Based on the above contents, the steps for attacking poermuation are briefly summarized as follows:

- **Step 1.** Select some special ordinary images and obtain their corresponding cipher images to determine the permutation matrix PM;
- **Step 2.** Use the permutation matrix PM to restore the original images from the permuted images.

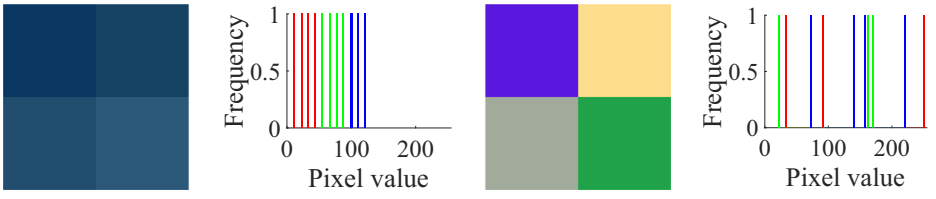
3.4 The proposed chosen-plain attack method

After the above discussion, the CIEA-IOCM cannot resist the attack method proposed in this paper. The flow chart of the attack method is shown in Fig. 4. The specific steps based on the chosen-plaintext attack are: First, obtain the equivalent diffusion matrix CM based on the method in Section 3.2. Secondly, the permutation matrix PM is obtained by the method in Section 3.3. Finally, use the equivalent keys to restore the original image.

In addition, the complexity required for the attack method is discussed here. In terms of data complexity, for a color image with a size of $M \times N \times 3$ the number of selected ordinary images required for decryption, diffusion and permutation is 1 and $\lceil \log_2(M \times 24N) \rceil$. Therefore, the total data complexity required is $O(1 + \lceil \log_2(M \times 24N) \rceil)$.

4 Experimental verifications and discussions

In order to verify our security analysis, the algorithm steps of CIEA-IOCM strictly follow Ref. [17]. We use MATLAB r2018b to perform simulation verification on a proposed image cryptography system based on PC (personal computer). The running PC is installed with Windows 10 64-bit OS (operating system), Intel (R) Core (TM) i5-8265U CPU @ 1.60GHz



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

Fig. 5 A pair of plain image and cipher image of size $2 \times 2 \times 3$: a) plain image I; b) the histogram of I; c) cipher image C; d) the histogram of C

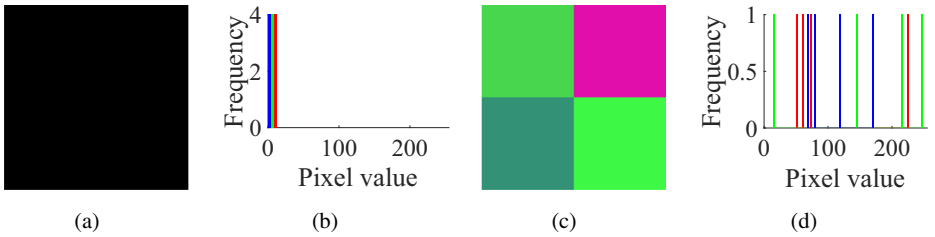


Fig. 6 The all-zero chosen plain image I^0 and its corresponding cipher image C^0 of size 2×2×3: a) I^0; b) the histogram of I^0; c) C^0; d) the histogram of C^0

and 8GB memory. We select some typical images listed in Table 1 for experiments. In (3), we set the experimental key parameters as: the initial value of chaotic mapping $x_0 = 0.34$, the control parameters $u = 2.56$, $k = 12$, $N_0 = 1000$, $kd = 654321$ and $rp = 1000$.

- **Case 1.** Breaking CIEA-IOCM with an image of size $2 \times 2 \times 3$:
  To better illustrate the attack process, we start with a very simple image of a size of 2×2×3. A pair of the given target plain image I and cipher images C is shown in Fig. 5a,c

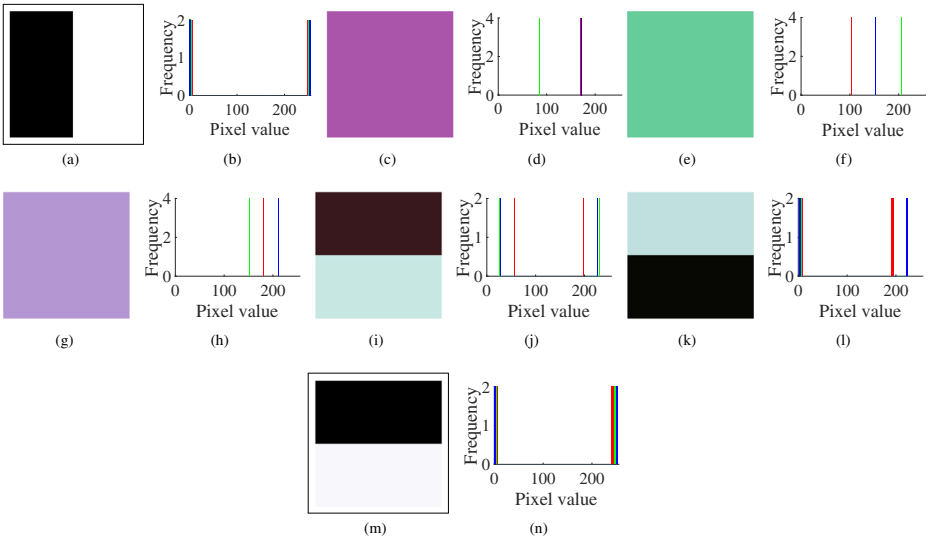


Fig. 7 Seven chosen plain images and their histograms of size $2 \times 2 \times 3$: a) $1^{\#}$ plain image; b) the histogram of a); c) $2^{\#}$ plain image; d) the histogram of c); e) $3^{\#}$ plain image; f) the histogram of e); g) $4^{\#}$ plain image; h) the histogram of g); i) $5^{\#}$ plain image; j) the histogram i); k) $6^{\#}$ plain image; l) the histogram of k); m) $7^{\#}$ plain image; n) the histogram of m)

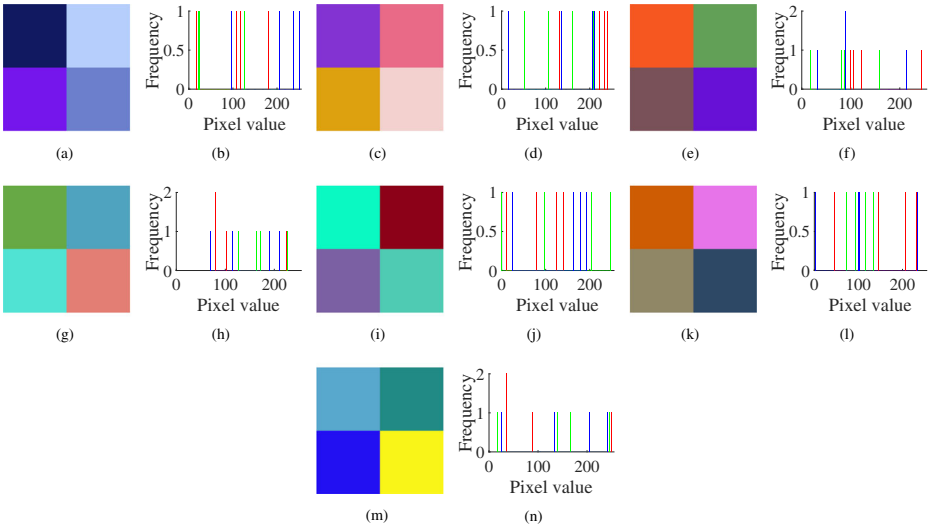


Fig. 8 Seven cipher images corresponding to seven chosen plain images and their histograms of size $2 \times 2 \times 3$: a) $1^{\#}$ cipher image; b) the histogram of a); c) $2^{\#}$ cipher image; d) the histogram of c); e) $3^{\#}$ cipher image; f) the histogram of e); g) $4^{\#}$ cipher image; h) the histogram of g); i) $5^{\#}$ cipher image; j) the histogram of i); k) $6^{\#}$ cipher image; l) the histogram of k); m) $7^{\#}$ cipher image; n) the histogram of m)

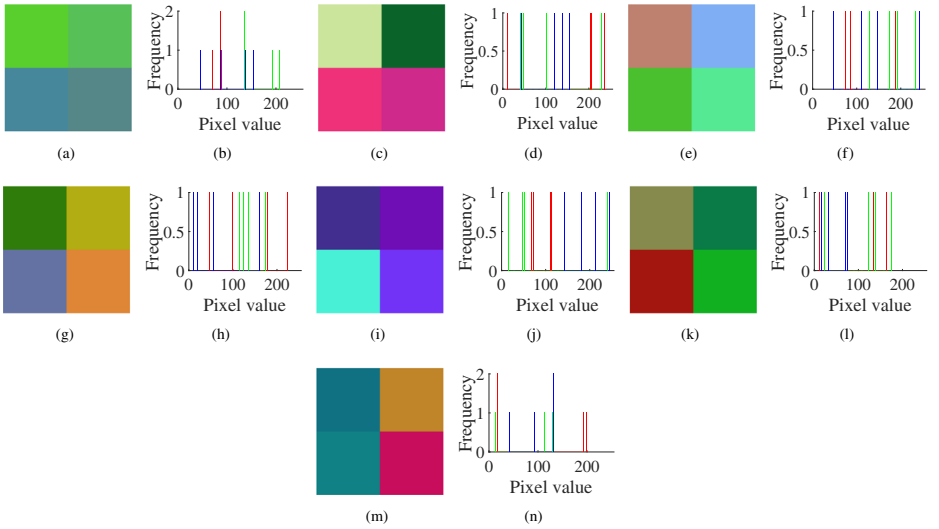


Fig. 9 Seven permuted images corresponding to seven chosen plain images and their histograms of size $2 \times 2 \times 3$: a) $1^{\#}$ permuted image; b) the histogram of a); c) $2^{\#}$ permuted image; d) the histogram of c); e) $3^{\#}$ permuted image; f) the histogram of e); g) $4^{\#}$ permuted image; h) the histogram of g); i) $5^{\#}$ permuted Image; j) the histogram of i); k) $6^{\#}$ permuted image; I) the histogram of k); m) $7^{\#}$ permuted image; n) the histogram of m)

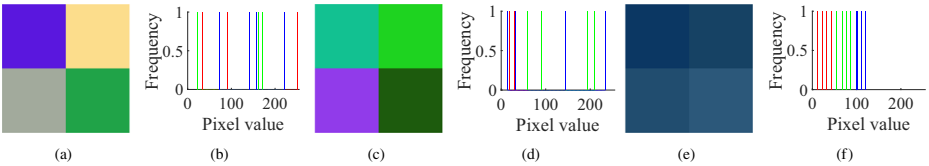


Fig. 10 A target cipher image, the permuted image, the original plain image and their histograms of size $2 \times 2 \times 3$: a) a target cipher image; b) the histogram of a); c) its permuted image; d) the histogram of c); e) its original plain image; f) the histogram of e)

respectively, and their histograms are shown in Fig. 5b,d respectively. Accordingly, the numerical matrices of I and C are:

$$\boldsymbol{IR} = \begin{bmatrix} 11 & 22 \\ 33 & 44 \end{bmatrix}; \boldsymbol{IG} = \begin{bmatrix} 55 & 66 \\ 77 & 88 \end{bmatrix}; \boldsymbol{IB} = \begin{bmatrix} 99 & 100 \\ 111 & 122 \end{bmatrix}$$

$$\boldsymbol{CR} = \begin{bmatrix} 90 & 252 \\ 162 & 32 \end{bmatrix}; \boldsymbol{CG} = \begin{bmatrix} 23 & 221 \\ 170 & 163 \end{bmatrix}; \boldsymbol{CB} = \begin{bmatrix} 222 & 140 \\ 156 & 72 \end{bmatrix}$$

Firstly, based on Step 1 in Section 3.2, choose the all-zero plain image I^0 shown in Fig. 6a and temporarily use the encryption machine of CIEA-IOCM, and then the corresponding cipher image C^0 is obtained, as shown in Fig. 6c. The all-zero image

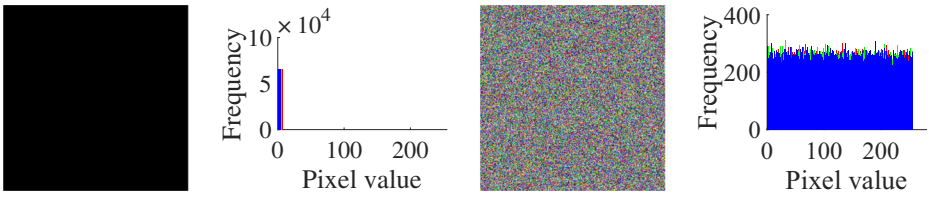


Fig. 11 The all-zero chosen plain image I^0, its corresponding cipher image C^0 and their histograms of size $256 \times 256 \times 3$: a) I^0; d) the histogram of a); c) C^0; d) the histogram of c)

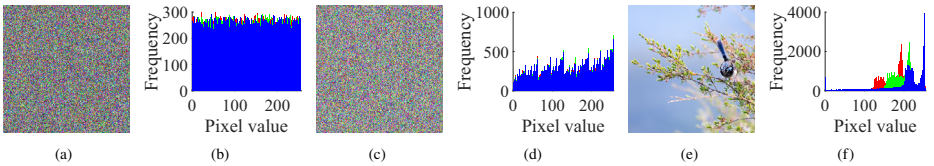


Fig. 12 The cipher image, permuted image, plain image and their histograms of size $256 \times 256 \times 3$: a) the cipher image; b) the histogram of a); c) its permuted image; d) the histogram of c); e) its plain image; f) the histogram of e)

I^0 and the corresponding cipher image C^0 and their histograms are shown in Fig. 6b,d respectively. Similarly, the numerical matrices of I^0 and C^0 are:

$$\boldsymbol{IR}^0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \boldsymbol{IG}^0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \boldsymbol{IB}^0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\boldsymbol{CR}^0 = \begin{bmatrix} 72 & 226 \\ 51 & 61 \end{bmatrix}; \boldsymbol{CG}^0 = \begin{bmatrix} 214 & 14 \\ 145 & 248 \end{bmatrix}; \boldsymbol{CB}^0 = \begin{bmatrix} 79 & 172 \\ 118 & 69 \end{bmatrix}$$

Secondly, the cipher image C^0 is decomposed at the bit-level in a method of Step 1 in Section 2.2 to obtain the one-dimensional image bit matrix C^0_{bit}. Then, according to the (11), we can conclude that $CM = C^0_{bit}$. Then, CM can be used to restore the permutation image shown in Fig. 10c from Fig. 10a.

Thirdly, following the Step 1 in Section 3.2, we choose some special attack images (shown in Figs. 7a-n) and get the corresponding cipher images (shown in Fig. 8a-n). Then, the equivalent diffusion matrix CM is used to obtain their corresponding permuated images (shown in Fig. 9a-n). After that, we can get PM as (15).

Finally, based on Step 2 in Section 3.3, we can restore the plain image shown in Fig. 10e from the permuted image shown in Fig. 10c with PM.

- **Case 2.** Breaking CIEA-IOCM with an iamge of size $256 \times 256 \times 3$:

  Firstly, based on Step 1 in Section 3.2, we choose the all-zero plain image I^0 and temporarily use the encryption machine of CIEA-IOCM, and then obtain the corresponding cipher image C^0. Their images are shown in Fig. 11a,c respectively, and their histograms are shown in Fig. 11b,d respectively. Then, the cipher image C^0 is decomposed at the bit-level in a method of Step 1 in Section 2.2 to obtain the one-dimensional image bit matrix C^0_{bit}. Then, according to (11), we could conclude that $CM = C^0_{bit}$.

  Secondly, we use the CM to restore the permuted image shown in Fig. 12c from the cipher image shown in Fig. 12a. Then we would obtain CM according to the (11). Thirdly, based on Step 1 in Section 3.3, construct some special attack images to obtain the permutation matrix PM. Finally, we restore the plain image shown in Fig. 12e from the permuted image shown in Fig. 12c with PM.

Without losing generality, we carried out experiments based on other images in different sizes. The experimental results are shown in Table 1 and Fig. 13. They all prove the effectiveness of our attack method. In addition, it can be seen from Table 1 that the proposed attack is effective. Taking the image of size $256 \times 256 \times 3$ as an example. When the time for encryption is 0.3115s, the time required for the corresponding attack is only 9.3985s. Even if the size of an image increases, the time required for attacking is still within an acceptable range. Therefore, it proves that our method is computationally feasible.

Last but not least, we verified the data complexity required for the attack. As described in Section 3.4, the total data complexity required for breaking the CIEA-IOCM is $O(1 +$

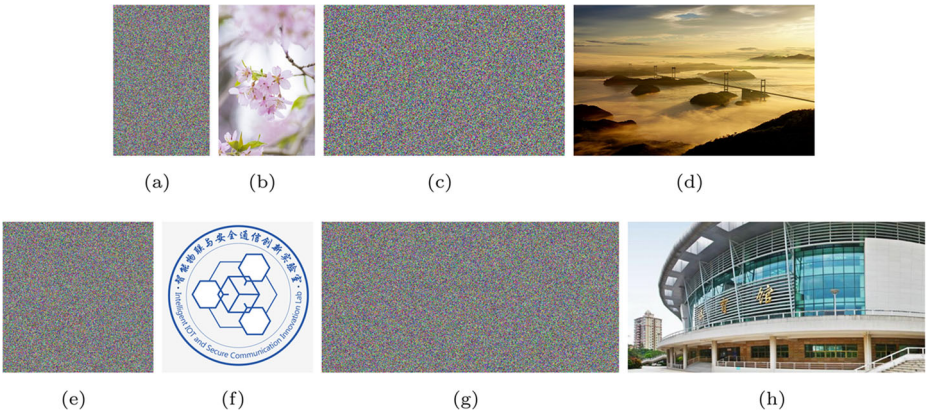


Fig. 13 Attacking results with four images of size 400×256×3, 320×512×3, 400×400×3 and 380×750×3 respectively: a) cipher image of size 400×256×3; b) plain image of a); c) cipher image of size 320×512× 3; d) plain image of c); e) cipher image of size 400 × 400 × 3; f) plain image of e); g) cipher image of size 380 × 750 × 3; h) plain image of g)

$\lceil \log_2(M \times 24N) \rceil$). In our experiment of chosen-plaintext attack, the numbers of attacking images with a size in $2 \times 2 \times 3$ and $400 \times 256 \times 3$ are 8 and 23, respectively. And for sizes of $256 \times 256 \times 3$ and $320 \times 512 \times 3$, the numbers of attacking images required are 22 and 23, respectively. Therefore, the experimental verification is consistent with the theoretical calculation.

5 Conclusion

In this paper, the security performance of a color image encryption algorithm named CIEA-IOCM using an improved one-dimensional chaotic map is analyzed. From the perspective of cryptanalysis, The encryption analysis which we conducted reveals that the

Table 1 The time required for breaking the improved 1D chaotic map by our proposed attack method (unit: second)

Images	Sizes	Encrytion	Attacking diffusion	Attacking permutation	Totol		
		time	Step 1	Step 1	Step 2	Attacking time	Data complexity
Fig. 10(e)	2 × 2 × 3	0.0051	0.5242	0.0253	0.0038	0.5533	$O(8)$
Fig. 12(e)	256 × 256 × 3	0.3115	0.5242	8.4357	0.4386	9.3985	$O(22)$
Fig. 13(b)	400 × 256 × 3	0.4804	0.8432	13.8091	0.7233	15.3756	$O(23)$
Fig. 13(d)	320 × 512 × 3	0.7799	1.3712	22.8973	1.2234	25.4919	$O(23)$
Fig. 13(f)	400 × 400 × 3	0.7233	1.2996	21.3472	1.0872	23.7340	$O(23)$
Fig. 13(h)	380 × 750 × 3	1.3360	2.2967	40.5493	2.0553	44.9013	$O(24)$

original algorithm has defects in security. The process of permutation and diffusion are not related to plaintext, and the algorithm has the equivalent keys. Therefore, this paper presents an approach to decode the original algorithm by obtaining the equivalent diffusion and permutation key respectively, based on the chosen-plaintext attack. Additionally, for the color images with the size $M \times N \times 3$, the data complexity for decrypting is $O(1 + \lceil \log_2(M \times 24N) \rceil)$. The cryptanalysis in this paper would provide some references for chaotic encryption designers to advance the security performance of the algorithm.

Funding This work was supported in part by the National Science Foundation of China under Grant 62071088, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011717, and in part by Project for Zhongshan Science and Technology under Grant 2021B2062.

Declarations

Conflict of Interests The authors declare no conflict of interest.

References

1. Ali TS, Ali R (2020) A new chaos based color image encryption algorithm using permutation substitution and boolean operation. Multimed Tools Appl 79(27-28):19853–19873
2. Aqeel-ur-Rehman, Liao X, Hahsmi MA, Haider R (2018) An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using dna and chaos. Optik 153:117–134
3. Bouteghrine B, Tanougast C, Sadoudi S (2021) Novel image encryption algorithm based on new 3-d chaos map. Multimed Tools Appl 80(17):25583–25605
4. Cao C, Sun K, Liu W (2018) A novel bit-level image encryption algorithm based on 2d-licm hyperchaotic map. Signal Process 143:122–133
5. Chai X, Gan Z, Yuan K, Chen Y, Liu X (2019) A novel image encryption scheme based on dna sequence operations and chaotic systems. Neural Comput Appl 31(1):219–237
6. Chai X, Zheng X, Gan Z, Han D, Chen Y (2018) An image encryption algorithm based on chaotic system and compressive sensing. Signal Process 148:124–144
7. Dai J-Y, Ma Y, Zhou N-R (2021) Quantum multi-image compression-encryption scheme based on quantum discrete cosine transform and 4d hyper-chaotic henon map. Quantum Inf Process 20(7)
8. Diaconu A-V (2016) Circular inter-intra pixels bit-level permutation and chaos-based image encryption. Inf Sci 355-356:314–327
9. Farah MAB, Guesmi R, Kachouri A, Samet M (2020) A novel chaos based optical image encryption using fractional fourier transform and dna sequence operation. Opt Laser Technol 121:105777
10. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurc Chaos Appl Sci Eng 8(6):1259–1284
11. Fu C, Huang J-B, Wang N-N, Hou Q-B, Lei W-M (2014) A symmetric chaos-based image cipher with an improved bit-level permutation strategy. Entropy 16(2):770–788
12. Kamrani A, Zenkouar K, Najah S (2020) A new set of image encryption algorithms based on discrete orthogonal moments and chaos theory. Multimed Tools Appl 79(27-28):20263–79
13. Kengnou Telem AN, Fotsin HB, Kengne J (2021) Image encryption algorithm based on dynamic dna coding operations and 3d chaotic systems. Multimed Tools Appl 80(12):19011–19041
14. Li Y, Wang C, Chen H (2017) A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Opt Lasers Eng 90:238–246
15. Liu H, Wang X (2011) Color image encryption using spatial bit-level permutation and high-dimension chaotic system. Opt Commun 284(16-17):3895–3903
16. Matthews R (1989) On the derivation of a "chaotic" encryption algorithm. Cryptologia 13(1):29–42
17. Pak C, An K, Jang P, Kim J, Kim S (2019) A novel bit-level color image encryption using improved 1d chaotic map. Multimed Tools Appl 78(9)
18. Shafique A, Shahid J (2018) Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. Eur Phys J Plus 133(8):331
19. Shannon CE (1949) Communication theory of secrecy systems. Bell Syst Tech J 28(4):655–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x

20. Sun S (2018) A novel hyperchaotic image encryption scheme based on dna encoding, pixel-level scrambling and bit-level scrambling. IEEE Photonics J 10(2):7201714
21. Teng L, Wang X, Meng J (2018) A chaotic color image encryption using integrated bit-level permutation. Multimed Tools Appl 77(6):6883–6896
22. Valli D, Ganesan K (2017) Chaos based video encryption using maps and ikeda time delay system. Eur Phys J Plus 132(12):542
23. Wang Y, Li X-W, Wang Q-H (2021) Integral imaging based optical image encryption using ca-dna algorithm. IEEE Photo J 13(2):1–12
24. Wang L, Ran Q, Ma J (2020) Double quantum color images encryption scheme based on dqrci. Multimed Tools Appl 79(9-10):6661–6687
25. Wang S, Wang C, Xu C (2020) An image encryption algorithm based on a hidden attractor chaos system and the knuth–durstenfeld algorithm. Opt Lasers Eng 128:105995
26. Wang Y, Zhao Y, Zhou Q, Lin Z (2018) Image encryption using partitioned cellular automata. Neurocomputing 275:1318–1332
27. Wen H, Yu S (2019) Cryptanalysis of an image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. Eur Phys J Plus 134(7):337
28. Wen H, Xu J, Liao Y, Chen R, Shen D, Wen L, Shi Y, Lin Q, Liang Z, Zhang S, Liu Y, Huo A, Li T, Cai C, Wen J, Zhang C (2021) A security-enhanced image communication scheme using cellular neural network. Entropy 23(8)
29. Wen H, Zhang C, Chen P, Chen R, Xu J, Liao Y, Liang Z, Shen D, Zhou L, Ke J (2021) A quantum chaotic image cryptosystem and its application in iot secure communication. IEEE Access 9:20481–20492
30. Wen H, Zhang C, Huang L, Ke J, Xiong D (2021) Security analysis of a color image encryption algorithm using a fractional-order chaos. Entropy 23(2)
31. Wen H, Yu S, Luuml J (2019) Breaking an image encryption algorithm based on dna encoding and spatiotemporal chaos. Entropy 21(3):246
32. Wu J, Liao X, Yang B (2018) Image encryption using 2d hénon-sine map and dna approach. Signal Process 153:11–23
33. Xie EY, Li C, Yu S, Lu J (2017) On the cryptanalysis of fridrich's chaotic image encryption scheme. Signal Process 132:150–154
34. Xu L, Li Z, Li J, Hua W (2016) A novel bit-level image encryption algorithm based on chaotic maps. Opt Lasers Eng 78:17–25
35. Zhang Y-Q, Wang X-Y (2014) Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation. Nonlinear Dyn 77(3):687–698
36. Zhang LY, Liu Y, Wang C, Zhou J, Zhang Y, Chen G (2018) Improved known-plaintext attack to permutation-only multimedia ciphers. Inf Sci 430-431:228–239
37. Zhang Y, Xiao D (2014) An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. Commun Nonlinear Sci Numer Simul 19(1):74–82
38. Zhang Y, Xiao D (2014) An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. Commun Nonlinear Sci Numer Simul 19(1):74–82
39. Zhu Z-L, Zhang W, Wong K-W, Yu H (2011) A chaos-based symmetric image encryption scheme using a bit-level permutation. Inf Sci 181(6):1171–1186