entropy

MDPI

Article

Security Analysis of the Image Encryption Algorithm Based on a Two-Dimensional Infinite Collapse Map

Guanwen Shi, Simin Yu and Qianxue Wang *

College of Automation, Guangdong University of Technology, Guangzhou 510006, China; sgwxuexi@126.com (G.S.); siminyu@163.com (S.Y.)

* Correspondence: wangqianxue@gdut.edu.cn

Abstract: This paper analyzes the security of the image encryption algorithm based on a two-dimensional (2D) infinite collapse map. The encryption algorithm adopts a permutation–diffusion structure and can perform two or more rounds to achieve a higher level of security. By cryptanalysis, it is found that the original diffusion process can be split into a permutation–diffusion structure, which comes after the original permutation, so these two permutations can be merged into one. Then, some theorems about round-down operation are summarized, and the encryption and decryption equations in the diffusion process are deduced and simplified accordingly. Since the chaotic sequences used in encryption algorithm are independent of the plaintext and ciphertext, there are equivalent keys. The original encryption algorithm with single-round, two-round, and multi-round of permutation–diffusion processes is cracked, and the data complexity of the cryptanalysis attacks is analyzed. Numerical simulation is carried out by MATLAB, and the experimental results and theoretical analysis show the effectiveness of the cryptanalysis attacks. Finally, some suggestions for improvement are given to overcome the shortcomings of the original encryption algorithm.

Keywords: chaotic image encryption; cryptanalysis; 2D infinite collapse map; equivalent key

check for updates

Citation: Shi, G.; Yu, S.; Wang, Q. Security Analysis of the Image Encryption Algorithm Based on a Two-Dimensional Infinite Collapse Map. *Entropy* **2022**, *24*, 1023. https://doi.org/10.3390/e24081023

Academic Editors: Xiaowei Li, Jian-Zhong Li and Yu Zhao

Received: 13 June 2022

Accepted: 22 July 2022

Published: 25 July 2022

1. Introduction

Advances in information and network technology have facilitated the rapid development of the Internet in providing the technical foundation, and the Internet is deeply integrated into all aspects of human life. Accompanying this is a variety of data forms and massive amounts of data generated every day. Since these data are closely linked with user information, their protection is particularly important. Digital image data are an important carrier of information, and has occupied a large part in the process of network transmission. Encrypting images is an important means to ensure image security.

Image data have the characteristics of strong correlation between pixels, high data redundancy, and large amount of data. The traditional text encryption algorithms such as DES and AES are not suitable for image encryption. In recent years, image encryption based on chaotic systems [1–7], cellular automata [8–12], DNA encoding [13–15], bit plane decomposition [16–21], and elliptic curve [22–26] is the mainstream of cryptography. Due to the significant properties of unpredictability, ergodicity and initial state sensitivity, the chaotic system becomes a good choice for encryption [27]. However, the chaotic sequence is transformed to a bit sequence to encrypt the plaintext in most chaotic image encryptions. The security of the encryption is thus determined by the properties of the bit sequence. Moreover, the essential reason for the chaotic cryptosystem easily existing equivalent keys is that the encryption process is independent of plaintext and/or ciphertext. In addition, elliptic curve cryptography is capable of providing high security than to other cryptosystems with the same key size because it is more complicated and requires a deeper mathematical understanding; it is more susceptible to errors which diminishes its security.

Since Matthews proposed a generalized logistic map and used it to generate pseudo-random numbers for data encryption [28], a large number of scholars have poured into using chaotic systems to design and implement novel image encryption schemes. In 1998, Fridrich [29] first proposed a chaotic image encryption scheme with multi-round of permutation–diffusion processes, which gradually became the main operation in chaotic image encryption algorithms. In 2015, Simin Yu reviewed the current situation and existing problems of the theory and application of chaotic cryptography, the literature [30] focused on the progress of high-dimensional chaotic cryptography and its application in multimedia secure communication and hardware implementation technology. In 2018, Özkaynak reported various chaotic image encryption algorithms proposed in the past 20 years. The chaotic systems, diffusion operations, and analysis methods commonly used in chaotic image encryption algorithms are classified and summarized in detail [31]. Overall, the chaotic encryption algorithm with a multi-round of permutation–diffusion processes offers cryptographic properties better than those with a single-round of permutation–diffusion processes, and it can resist against the chosen-plaintext attacks.

Cryptography and cryptanalysis are the unity of opposites, and they promote each other and develop together. Through cryptanalysis, the defects of cryptographic algorithm can be pointed out and the suggestions for improvement are given. Cryptanalysis is based on the Kerchhoff's principle; a cryptographic system should be secure even if everything about the system, except the key, is public knowledge. The attacker can get the plaintext or even the encryption key through the obtained plaintext/ciphertext pair. In cryptology, the basic models are named after the generally defined attacks such as ciphertext-only attack, known-plaintext attack, chosen-plaintext attack, and chosen-ciphertext attack. Many analysis methods can be classified into the above four methods. In recent years, linear attack and differential attack [32] have been proposed one after another, which have a great impact on cryptanalysis. Many new analysis methods are variants of these two methods [33].

At present, there are some analytical articles on a multi-round image encryption algorithm. In 2010, Solak et al. proposed a chosen-ciphertext attack on the Fridrich's scheme for the first time [34]. Some bases for further optimizing attack on the Fridrich's scheme and its variants are provided in [35]. In 2015, Chen et al. analyzed an encryption algorithm with a multi-round of permutation–diffusion structure [36], and proposed a differential cryptanalysis method for two-round and multi-round [37]. However, due to the special permutation operation adopted by the original encryption algorithm, the analysis for multi-round is not universal; in 2016, they proposed a method of chosen-ciphertext attack, and verified the adaptability of this attack method by analyzing several common diffusion equations [38]. In 2021, a multi-round chaotic image encryption algorithm was analyzed in [39]. The original encryption algorithm adopts multiple permutations and one diffusion, and repeats them multiple rounds. Multiple consecutive permutations are equivalent to one permutation. Since the diffusion operation only uses XOR without ciphertext feedback, the diffusion part can be separated from the permutation part. Therefore, it can be cracked by simplifying it into one round of permutation–diffusion. In the same year, Chen et al. mathematically summarized and expressed a class of chaotic image cryptosystems based on a multi-round of permutation–diffusion structure [37,38], and proposed a chosen-ciphertext attack method for this kind of encryption algorithm [40,41]. It is noted that the cryptanalysis algorithms in the existing literature are mainly aimed at the single-round encryption and some multi-round encryptions, which also can be directly equivalent to single-round encryption after simplification.

In this paper, a security analysis of the image encryption algorithm based on a 2D infinite collapse map proposed in [42] is carried out. According to the analysis, the encryption algorithm has one permutation operation in the diffusion process. Therefore, its structure is actually a permutation–permutation–diffusion structure, and two permutation operations can be equivalent to one permutation operation. In addition, this paper deduces the rules of round-down operation, and then gives the correct diffusion decryption equation. Since the chaotic sequences used in the encryption algorithm are independent of the plaintext and ciphertext, there are equivalent keys. This paper analyzes and discusses the single-round, two-round and multi-round situations, provides the attack complexity, and gives the corresponding improvement suggestions to overcome the shortcomings of the original encryption algorithm. The main advantage of this paper is that a detailed security analysis of a more complex multi-round encryption algorithm is carried out, and the main difference between this multi-round encryption and the previous multi-round encryption methods is that the multi-round encryption cannot be directly equivalent to a single-round of encryption. Therefore, the cryptanalysis methods in the existing literature cannot be directly used to crack this multi-round encryption algorithm.

The remainder of this article is organized as follows: Some definitions and related theorems are provided in Section 2. Section 3 presents the detail of the original encryption algorithm, and gives the correct decryption equation. An analysis of the encryption algorithm is demonstrated in detail in Section 4. Section 5 mainly introduces the numerical simulation experiments carried out by MATLAB. The experimental results verify the correctness of the cryptanalysis, and at the same time, the complexity of the deciphering algorithms is given, and corresponding improvement measures are proposed to overcome the shortcomings of the original encryption algorithm. The last section concludes the article.

2. Some Definitions and Related Theorems

In order to better analyze the original encryption algorithm, it is first necessary to simplify the original algorithm. According to the formula used in the original algorithm, some preliminaries are given to aid the subsequent theoretical analysis. The definitions and properties of round-down operation $\lfloor \cdot \rfloor$, the operation $\{\cdot\}$ for finding the fractional part of a real number, and the modulus operator are introduced, and three theorems about these operations are deduced in this section.

Definition 1 ([43])**.** *The largest integer of a real number a is recorded as $\lfloor a \rfloor$, which is the largest integer less than or equal to a, that is, $\lfloor a \rfloor$ is the integer satisfying $\lfloor a \rfloor \leq a < \lfloor a \rfloor + 1$.*

Definition 2 ([43])**.** *The fractional part of the real number a is recorded as $\{a\}$, which is the difference between a and $\lfloor a \rfloor$, that is, $\{a\} = a - \lfloor a \rfloor$.*

Property 1. $a = \lfloor a \rfloor + \{a\}, 0 \leq \{a\} < 1.$

Property 2. $\lfloor n + a \rfloor = n + \lfloor a \rfloor, \{n + a\} = \{a\}, n \in \mathbb{Z}.$

Theorem 1. *For any real numbers a and b, there are*

$$\lfloor a + b \rfloor = \begin{cases} \lfloor a \rfloor + \lfloor b \rfloor & 0 \leq \{a\} + \{b\} < 1, \\ \lfloor a \rfloor + \lfloor b \rfloor + 1 & 1 \leq \{a\} + \{b\} < 2. \end{cases}$$

Proof.

$$\begin{aligned} \lfloor a + b \rfloor &= \lfloor (\lfloor a \rfloor + \{a\}) + (\lfloor b \rfloor + \{b\}) \rfloor && (\text{Property } 1) \\ &= \lfloor \lfloor a \rfloor + \lfloor b \rfloor + (\{a\} + \{b\}) \rfloor \\ &= \lfloor a \rfloor + \lfloor b \rfloor + \lfloor \{a\} + \{b\} \rfloor. && (\lfloor a \rfloor + \lfloor b \rfloor \in \mathbb{Z}, \text{Property } 2). \end{aligned}$$

From Property 1, we know $0 \leq \{a\} < 1, 0 \leq \{b\} < 1$, so $0 \leq \{a\} + \{b\} < 2$.
When $0 \leq \{a\} + \{b\} < 1$, $\lfloor \{a\} + \{b\} \rfloor = 0$, then $\lfloor a + b \rfloor = \lfloor a \rfloor + \lfloor b \rfloor$.
When $1 \leq \{a\} + \{b\} < 2$, $\lfloor \{a\} + \{b\} \rfloor = 1$, then $\lfloor a + b \rfloor = \lfloor a \rfloor + \lfloor b \rfloor + 1$.
□

Theorem 2. *For any real numbers a and b, there are*

$$\lfloor a - b \rfloor = \begin{cases} \lfloor a \rfloor - \lfloor b \rfloor & 0 \leq \{a\} - \{b\} < 1, \\ \lfloor a \rfloor - \lfloor b \rfloor - 1 & -1 < \{a\} - \{b\} < 0. \end{cases}$$

Proof.

$$\begin{aligned}\lfloor a-b\rfloor &= \lfloor(\lfloor a\rfloor+\{a\})-(\lfloor b\rfloor+\{b\})\rfloor && (\text{Property } 1)\\ &= \lfloor\lfloor a\rfloor-\lfloor b\rfloor+(\{a\}-\{b\})\rfloor\\ &= \lfloor a\rfloor-\lfloor b\rfloor+\lfloor\{a\}-\{b\}\rfloor. && (\lfloor a\rfloor-\lfloor b\rfloor\in\mathbb{Z}, \text{Property } 2)\end{aligned}$$

From Property 1, we know $0 \le \{a\} < 1, 0 \le \{b\} < 1$, so $-1 < \{a\} - \{b\} < 1$.
When $0 \le \{a\} - \{b\} < 1$, $\lfloor\{a\} - \{b\}\rfloor = 0$, then $\lfloor a-b\rfloor = \lfloor a\rfloor - \lfloor b\rfloor$.
When $-1 < \{a\} + \{b\} < 0$, $\lfloor\{a\} - \{b\}\rfloor = -1$, then $\lfloor a-b\rfloor = \lfloor a\rfloor - \lfloor b\rfloor - 1$.
□

Definition 3 ([44])**.** *The modular operation returns the remainder after a real number is divided by a positive integer, and often abbreviated as* mod*:*

Property 3.

$$(a \bmod 256) \bmod 256 = a \bmod 256, a \in R.$$

Property 4.

$$(a+b) \bmod 256 = ((a \bmod 256) + (b \bmod 256)) \bmod 256, a, b \in R.$$

Theorem 3.

$$\lfloor a \bmod 256\rfloor = \lfloor a\rfloor \bmod 256, a \in R.$$

Proof. Assuming $b = a \bmod 256$, the corresponding inverse operation is $a = 256 \times k + b$, where $a, b \in R$, $k \in \mathbb{Z}$ and $0 \le b < 256$, so $\lfloor a \bmod 256\rfloor = \lfloor b\rfloor$ and

$$\begin{aligned}\lfloor a\rfloor \bmod 256 &= \lfloor 256\times k+b\rfloor \bmod 256\\ &= (256\times k+\lfloor b\rfloor) \bmod 256 && (256\times k\in\mathbb{Z}, \textit{Property } 2)\\ &= \lfloor b\rfloor. && (0\le\lfloor b\rfloor<256, \text{Definition } 3)\end{aligned}$$

$\lfloor a \bmod 256\rfloor = \lfloor b\rfloor = \lfloor a\rfloor \bmod 256$ is proved. □

3. Description of the Original Encryption Algorithm

In this section, the chaotic map used in [42] is first introduced, and then the original encryption algorithm is described in detail.

3.1. Two-Dimensional Infinite Collapse Map (2D-ICM)

The chaotic system 2D-ICM used in the original encryption algorithm is a two-dimensional infinite collapse map obtained by integrating two one-dimensional infinite collapse maps with different parameters [42], and its iterative equation is

$$\begin{cases} x_{n+1} = \sin\left(\frac{a}{y_n}\right) \cdot \sin\left(\frac{b}{x_n}\right), \\ y_{n+1} = \sin\left(\frac{a}{x_n}\right) \cdot \sin\left(\frac{b}{y_n}\right), \end{cases} \tag{1}$$

where the control parameters a and b are real numbers, $a \neq 0$, $b \neq 0$, and the initial states are recorded as x_0, y_0.

3.2. 2D-ICM Based Image Encryption Algorithm (ICMIE)

According to [42], it proposed a new image encryption algorithm based on 2D-ICM and named it ICMIE. The original algorithm ICMIE is described as follows:

(1) Key parameters

There are seven key parameters in the original algorithm. The key K is expressed as $a_0, b_0, x, y, T, C_1, C_2$, and the first five parameters a_0, b_0, x, y, T are 40-bit binary representation. Assuming that the 40-bit binary is $(s_1 s_2 \cdots s_{40})_2$, the IEEE754 format

$$d = \frac{\sum_{i=1}^{40} s_i 2^{40-i}}{2^{40}} \tag{2}$$

is adopted to convert a_0, b_0, x, y, T into decimal numbers in $[0, 1)$, then C_1 and C_2 are positive integers represented by 20-bit binary, and they are converted into decimal numbers directly. Substitute the converted decimal $a_0, b_0, x, y, T, C_1, C_2$ into the following equation:

$$\begin{cases} a = (a_0 + T \times C_1) \bmod 5 + 16, \\ b = (b_0 + T \times C_2) \bmod 5 + 16, \\ x_0 = (x + T \times C_1) \bmod 2 - 1, \\ y_0 = (y + T \times C_2) \bmod 2 - 1. \end{cases} \tag{3}$$

The initial conditions a, b, x_0, y_0 of 2D-ICM can be obtained.

(2) Encryption process

The original algorithm divides the encryption process into permutation and diffusion, and then performs two or more rounds of permutation and diffusion as a whole. In fact, the diffusion process of the original algorithm also includes a permutation operation. In order to distinguish, the first permutation operation is named permutation 1 and the second permutation operation is named permutation 2. The grayscale image P of $M \times N$ is encrypted, and the ciphertext image C of the same size is finally generated. The overall encryption process is shown in Figure 1.

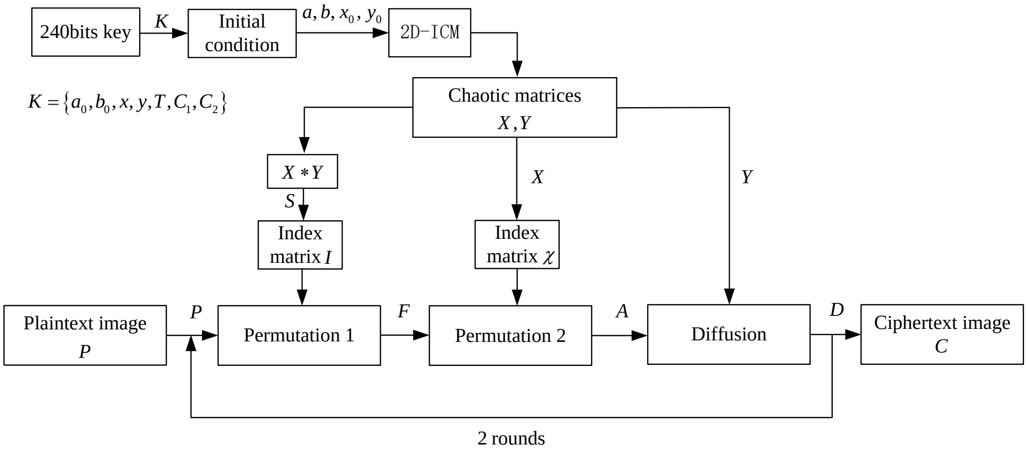


Figure 1. The algorithmic structure for ICMIE.

The specific encryption steps are described as follows:

(1) Permutation

First, two chaotic matrices X and Y of $M \times N$ are generated by 2D-ICM. The matrix S is combined into a single matrix $S = X * Y$ by multiplying the corresponding elements of X and Y. The index matrix I is composed of the position of each element in the original matrix S after sorting S in ascending order. Then, the pixel positions of the plaintext image P are rearranged by using the index matrix I to obtain the permutation 1 image F.

(2) Diffusion

First, the index matrix χ is composed of the corresponding positions of all elements of the chaotic matrix X in its ascending order. Then, the pixel positions of the permutation

1 image F are arranged again by the index matrix χ to obtain a new permutation 2 image A. Finally, the pixel values of the diffusion image D are obtained by the following method:

$$d_i = \begin{cases} \lfloor (a_i + a_{M\times N} + |y_i| \times (2^{31}-1)) \bmod 256 \rfloor & \text{if } i = 1, \\ \lfloor (a_i + d_{i-1} + |y_i| \times (2^{31}-1)) \bmod 256 \rfloor & \text{if } i \in [2, M\times N], \end{cases} \tag{4}$$

where $i = 1, 2, \ldots, M \times N$, then d_i, a_i and y_i are the pixel values of the i-th element of the diffusion image D, the permutation 2 image A and the chaotic matrix Y according to the raster scan order, respectively.

(3) Repeat steps (1) and (2) to achieve multi-round encryption.

(3) Decryption process

Usually, the decryption process is the inverse of the image encryption process. Using the correct key to generate the chaotic matrices X and Y, the decryption process of ICMIE will alternately perform the inverse diffusion and inverse permutation in two rounds or multiple rounds, and then obtain the recovered image. The decryption equation in the diffusion process is incorrect. When $i \in [2, M \times N]$, according to the encryption Equation (4),

$$\begin{aligned}
d_i &= \left\lfloor \left(a_i + d_{i-1} + |y_i| \times \left(2^{31}-1\right)\right) \bmod 256 \right\rfloor, \\
d_i &= \left\lfloor \left(a_i + d_{i-1} + |y_i| \times \left(2^{31}-1\right)\right)\right\rfloor \bmod 256, && \text{(Theorem 3)} \\
d_i + 256 \times k_i &= \left\lfloor \left(a_i + d_{i-1} + |y_i| \times \left(2^{31}-1\right)\right)\right\rfloor, && \text{(Definition 3)} \\
d_i + 256 \times k_i &= a_i + d_{i-1} + \left\lfloor \left(|y_i| \times \left(2^{31}-1\right)\right)\right\rfloor, && \text{(Property 2)} \\
a_i &= d_i + 256 \times k_i - d_{i-1} - \left\lfloor \left(|y_i| \times \left(2^{31}-1\right)\right)\right\rfloor, \\
a_i \bmod 256 &= \left(d_i + 256 \times k_i - d_{i-1} - \left\lfloor \left(|y_i| \times \left(2^{31}-1\right)\right)\right\rfloor\right) \bmod 256, \\
a_i &= \left(\left(d_i - d_{i-1} - \left\lfloor \left(|y_i| \times \left(2^{31}-1\right)\right)\right\rfloor\right) \bmod 256\right) \bmod 256, && \text{(Property 4)} \\
a_i &= \left(d_i - d_{i-1} - \left\lfloor \left(|y_i| \times \left(2^{31}-1\right)\right)\right\rfloor\right) \bmod 256, && \text{(Property 3)} \\
a_i &= \left\lfloor d_i - d_{i-1} - \left(|y_i| \times \left(2^{31}-1\right)\right) + 1 \right\rfloor \bmod 256, && \text{(Theorem 2)}
\end{aligned}$$

where $k_i \in \mathbb{Z}(i = 2, 3, \cdots, M \times N)$. Similarly, a_i can be obtained when $i = 1$, and the correct decryption equation is finally derived as

$$a_i = \begin{cases} \lfloor (d_i - d_{i-1} - (|y_i| \times (2^{31}-1)) + 1) \bmod 256 \rfloor & \text{if } i \in [2, M\times N], \\ \lfloor (d_i - a_{M\times N} - (|y_i| \times (2^{31}-1)) + 1) \bmod 256 \rfloor & \text{if } i = 1. \end{cases} \tag{5}$$

Then, the pixel positions of the ciphertext image will be processed by inverse permutation. The original image is completely recovered.

4. Cryptanalysis

The generation process of the chaotic sequences and the encryption process of the original algorithm are independent of the plaintext and the ciphertext, so there are equivalent keys. Firstly, the core structure of the original encryption algorithm (i.e., the permutation–permutation–diffusion structure) is generalized, then the diffusion equation is isolated, and the two permutation processes are merged into one permutation. Next, the original encryption algorithm is analyzed in terms of single-round, two-round, and multi-round.

The original encryption algorithm can be summarized as a multi-round of permutation–diffusion processes as shown in Figure 2.

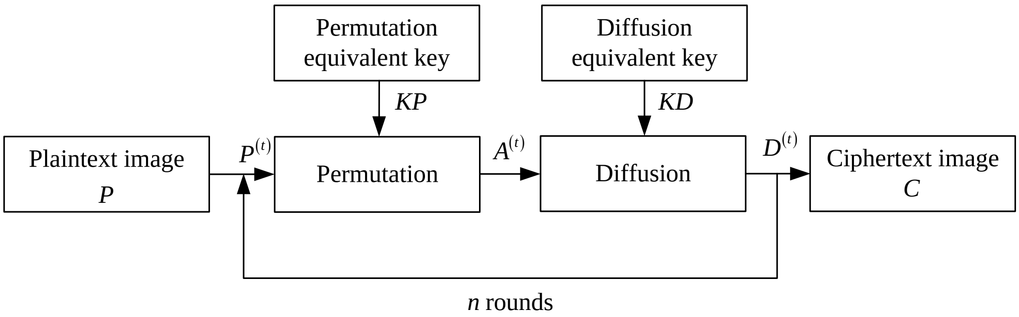


Figure 2. The block diagram of n-round chaotic image encryption.

In Figure 2, KP and KD represent the permutation equivalent key and the diffusion equivalent key, respectively. P represents plaintext image and C represents ciphertext image. n represents the total number of rounds of encryption, and

$$P^{(t)} = \begin{cases} P & t = 1, \\ D^{(t-1)} & t \in [2, n], \end{cases}$$

where $P^{(t)}$, $A^{(t)}$, and $D^{(t)}$ represent the plaintext image, the permutation image, and diffusion image encrypted in the t-th round, respectively. Taking the feedback apart, it can be shown in Figure 3.

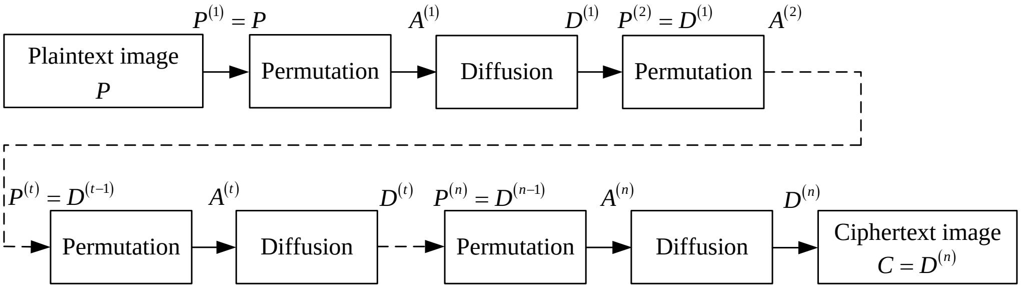


Figure 3. The block diagram of n-round chaotic image encryption without feedback.

In Figure 3, $P^{(t)} = D^{(t-1)}$, that is, the diffusion image encrypted in the previous round is the plaintext image encrypted in the subsequent round. For plaintext image P and ciphertext image C, there are $P^{(1)} = P$ and $C = D^{(n)}$.

For the convenience of the following discussion, some definitions are given here. $P_i^{(t)}, A_i^{(t)}, D_i^{(t)}$ respectively represent the i-th plaintext image, permutation image, and diffusion image in the t-th round. The size of the images discussed in this paper are all $M \times N$.

4.1. Simplification of ICMIE

Since the two permutations are independent of plaintext, they can be equivalent to one permutation operation. The equivalent key KP of the two permutation operations from plaintext image P to permutation image A can be obtained directly in one step. The diffusion equivalent key KD can be obtained from the permutation image A to the diffusion image D.

The original diffusion encryption Equation (4) is deduced and the following equation is obtained. When $i \in [2, M \times N]$, according to Theorem 1, there is

$$\begin{aligned} d_i &= \left\lfloor \left(a_i + d_{i-1} + |y_i| \times \left(2^{31} - 1\right) \right) \bmod 256 \right\rfloor \\ &= \left\lfloor a_i + d_{i-1} + |y_i| \times \left(2^{31} - 1\right) \right\rfloor \bmod 256 && \text{(Theorem 3)} \\ &= \left(a_i + d_{i-1} + \left\lfloor |y_i| \times \left(2^{31} - 1\right) \right\rfloor \right) \bmod 256 && \text{(Property 2)} \\ &= \left(a_i + d_{i-1} + \left\lfloor |y_i| \times \left(2^{31} - 1\right) \right\rfloor \bmod 256 \right) \bmod 256 && \text{(Property 4)} \\ &= (a_i + d_{i-1} + \hat{y}_i) \bmod 256, \end{aligned} \tag{6}$$

where $\hat{y}_i = \left(\lfloor |y_i| \times (2^{31} - 1) \rfloor\right) \bmod 256$.

Likewise, when $i = 1$,

$$d_1 = (a_1 + a_{M \times N} + \hat{y}_1) \bmod 256, \tag{7}$$

where $\hat{y}_1 = \left(\lfloor |y_1| \times (2^{31} - 1) \rfloor\right) \bmod 256$.

In order to facilitate the analysis, $\dot{+}$ is defined to represent the modular addition that is, the two elements are added and then modulo 256. Correspondingly, $\dot{-}$ represents the modular subtraction, that is, the two elements are subtracted and then modulo 256. From Equations (6) and (7),

$$d_i = \begin{cases} a_i \dot{+} a_{M \times N} \dot{+} \hat{y}_i & \text{if } i = 1, \\ a_i \dot{+} d_{i-1} \dot{+} \hat{y}_i & \text{if } i \in [2, M \times N], \end{cases} \tag{8}$$

where $\hat{y}_i = \left(\lfloor |y_i| \times (2^{31} - 1) \rfloor\right) \bmod 256$.

4.2. Security Analysis of Encryption in Single-Round

First, let P_0 be an all-zero image, then the pixel value will not be changed after permutation. Therefore, the element values of the permutation image A_0 are all 0. The image D_0 is obtained according to the encryption algorithm. According to Equation (8), one has

$$d_i = \begin{cases} \hat{y}_i & \text{if } i = 1, \\ d_{i-1} \dot{+} \hat{y}_i & \text{if } i \in [2, M \times N], \end{cases} \tag{9}$$

and

$$\hat{y}_i = \begin{cases} d_i & \text{if } i = 1, \\ d_i - d_{i-1} + 256k_i & \text{if } i \in [2, M \times N], \end{cases} \tag{10}$$

where $k_i \in \mathbb{Z}(i = 2, 3, \cdots, M \times N)$.

Because $\hat{y}_i$ and d_i perform modulo 256 operation,

$\hat{y}_i \in \{0, 1, 2, \cdots, 255\}, d_i - d_{i-1} \in \{-255, -254, \cdots, 254, 255\}(i = 2, 3, \cdots, M \times N)$, so

$$\hat{y}_i = \begin{cases} d_i & \text{if } i = 1, \\ d_i - d_{i-1} + 256 & \text{if } i \in [2, M \times N] \text{and} (d_i - d_{i-1}) < 0, \\ d_i - d_{i-1} & \text{if } i \in [2, M \times N] \text{and} (d_i - d_{i-1}) \geq 0. \end{cases} \tag{11}$$

In other words, substitute the pixel value of D_0 to obtain $\hat{y}_i$ by Equation (11), then make $kd_i = \hat{y}_i (i = 1, 2, \cdots, M \times N)$ to get the equivalent key $KD = kd_1 kd_2 \cdots kd_{M\times N}$. Next, according to Theorem 3, the diffusion decryption from Equation (5), one can further obtain

$$\begin{aligned} a_i &= \begin{cases} \lfloor d_i \dot{-} d_{i-1} \dot{-} \left(|y_i| \times \left(2^{31} - 1\right)\right) \dot{+} 1 \rfloor & \text{if } i \in [2, M \times N], \\ \lfloor d_i \dot{-} a_{M\times N} \dot{-} \left(|y_i| \times \left(2^{31} - 1\right)\right) \dot{+} 1 \rfloor & \text{if } i = 1. \end{cases} \\ &= \begin{cases} \lfloor (d_i \dot{-} d_{i-1} \dot{+} 1) \dot{-} \left(|y_i| \times \left(2^{31} - 1\right)\right) \rfloor & \text{if } i \in [2, M \times N], \\ \lfloor (d_i \dot{-} a_{M\times N} \dot{+} 1) \dot{-} \left(|y_i| \times \left(2^{31} - 1\right)\right) \rfloor & \text{if } i = 1. \end{cases} \end{aligned} \tag{12}$$

Because $d_i \dot{-} d_{i-1} \dot{+} 1, d_i \dot{-} a_{M\times N} \dot{+} 1 \in \mathbb{Z}$, it means $\{d_i \dot{-} d_{i-1} \dot{+} 1\} = 0$ and $\{d_i \dot{-} a_{M\times N} \dot{+} 1\} = 0$, then $\{d_i \dot{-} d_{i-1} \dot{+} 1\} \dot{-} \left\{|y_i| \times \left(2^{31} - 1\right)\right\} < 0$ and $\{d_i \dot{-} a_{M\times N} \dot{+} 1\} \dot{-} \left\{|y_i| \times \left(2^{31} - 1\right)\right\} < 0$. According to Theorem 2 and Property 2,

$$\begin{aligned} a_i &= \begin{cases} \lfloor d_i \dot{-} d_{i-1} \dot{+} 1 \rfloor \dot{-} \left\lfloor |y_i| \times \left(2^{31} - 1\right) \right\rfloor \dot{-} 1 & \text{if } i \in [2, M \times N], \\ \lfloor d_i \dot{-} a_{M\times N} \dot{+} 1 \rfloor \dot{-} \left\lfloor |y_i| \times \left(2^{31} - 1\right) \right\rfloor \dot{-} 1 & \text{if } i = 1, \end{cases} \\ &= \begin{cases} d_i \dot{-} d_{i-1} \dot{+} 1 \dot{-} \left\lfloor |y_i| \times \left(2^{31} - 1\right) \right\rfloor \dot{-} 1 & \text{if } i \in [2, M \times N], \\ d_i \dot{-} a_{M\times N} \dot{+} 1 \dot{-} \left\lfloor |y_i| \times \left(2^{31} - 1\right) \right\rfloor \dot{-} 1 & \text{if } i = 1, \end{cases} \\ &= \begin{cases} d_i \dot{-} d_{i-1} \dot{-} \hat{y}_i & \text{if } i \in [2, M \times N], \\ d_i \dot{-} a_{M\times N} \dot{-} \hat{y}_i & \text{if } i = 1, \end{cases} \end{aligned} \tag{13}$$

where $\hat{y}_i = \left(\left\lfloor |y_i| \times \left(2^{31} - 1\right) \right\rfloor\right) \bmod 256$.

The permutation image A corresponding to the ciphertext image C can be cracked by substituting the equivalent key KD (i.e., $\hat{y}_i = kd_i (i = 1, 2, \cdots, M \times N)$) obtained from all-zero plaintext and $d_i (i = 1, 2, \cdots, M \times N)$ according to Equation (13). Because the specific ciphertext C is known, then the diffusion image $D = C$, so $d_i (i = 1, 2, \cdots, M \times N)$ is known.

Since the two permutations are independent of the plaintext, they can be equivalent to one permutation, and the permutation operation only changes the coordinate position of the pixel without changing the pixel value, so that only the coordinate position of the pixel in the permutation image A is changed. Therefore, the equivalent permutation key KP can be solved by comparing the pixel pairs of the plaintext images and the permutation images. Next, the optimal chosen-plaintext attack is used [45], and the steps are as follows:

Step 1: Construct a data matrix U with the same size as the image P, u_j is the value of the j-th element of the matrix U obtained in raster scan order. The nonnegative integers $0, 1, \cdots, M \times N - 1$ are successively written into the data matrix U according to the raster scan order by $u_j = j - 1 (j = 1, 2, 3, \ldots, M \times N)$.

Step 2: Calculate the number of selected plaintexts $l = \lceil \log_{256}(M \times N) \rceil$, where $\lceil \cdot \rceil$ is the round-up operation. In addition, create l plaintext images $P_1, P_2, \cdots, P_l$.

Step 3: Use U to write the value into $P_1, P_2, \cdots, P_l$. The writing rule of the j-th element $p_{i,j}$ obtained from the i-th plaintext image in raster scan order is

$$p_{i,j} = \left\lfloor \left(u_j / 256^{i-1}\right) \% 256 \right\rfloor, \tag{14}$$

where $i = 1, 2, 3, \ldots, l$ and $j = 1, 2, 3, \ldots, M \times N$.

After constructing the plaintext through the above steps, l plaintext images $P_1, P_2, \cdots, P_l$ are successively input into the encryptor to obtain the corresponding ciphertext images $C_1, C_2, \cdots, C_l$, respectively. Then, according to the obtained equivalent diffusion key KD, inverse diffusion is carried out to obtain $A_1, A_2, \cdots, A_l$, respectively, and these images are combined into a data matrix V. The consolidation rule is

$$V = \sum_{i=1}^{l} \left(A_i \times 256^{i-1}\right), \tag{15}$$

where $i = 1,2,3,\ldots,l$. By comparing the position difference between the data matrix U and the data matrix V with the same pixel value, the equivalent permutation key KP used in permutation can be obtained.

4.3. Cryptanalysis of Two-Round Encryption

Two-round encryption is analyzed here by the combination of the differential attack and the chosen-plaintext attack.

Firstly, the encryption algorithm is deduced by differential analysis. According to Equation (8),

$$\begin{cases} d_1 & = a_{M\times N} \dotplus a_1 \dotplus \hat{y}_1, \\ d_2 & = a_{M\times N} \dotplus a_1 \dotplus a_2 \dotplus \hat{y}_1 \dotplus \hat{y}_2, \\ & \vdots \\ d_j & = a_{M\times N} \dotplus a_1 \dotplus a_2 \dotplus \ldots \dotplus a_j \dotplus \hat{y}_1 \dotplus \hat{y}_2 \dotplus \ldots \dotplus \hat{y}_j, \\ & \vdots \\ d_{M\times N} & = a_{M\times N} \dotplus a_1 \dotplus a_2 \dotplus \ldots \dotplus a_{M\times N} \dotplus \hat{y}_1 \dotplus \hat{y}_2 \dotplus \ldots \dotplus \hat{y}_{M\times N}. \end{cases} \tag{16}$$

Now use $a^{(t)}_{i,j}, d^{(t)}_{i,j}$ ($t = 1,2,3,\ldots,n$, $i = 0,1,2,\cdots$ and $j = 1,2,3,\ldots,M\times N$) to represent the j-th element of the i-th permutation image and diffusion image in the raster scan order in the t-th round of encryption, respectively. Then, the j-th element in raster scan order in two different diffusion images $D^{(t)}_k$ and $D^{(t)}_l$ encrypted in the t-th round can be expressed as

$$d^{(t)}_{k,j} = a^{(t)}_{k,M\times N} \dotplus a^{(t)}_{k,1} \dotplus a^{(t)}_{k,2} \dotplus \ldots \dotplus a^{(t)}_{k,j} \dotplus \hat{y}_1 \dotplus \hat{y}_2 \dotplus \ldots \dotplus \hat{y}_j \tag{17}$$

and

$$d^{(t)}_{l,j} = a^{(t)}_{l,M\times N} \dotplus a^{(t)}_{l,1} \dotplus a^{(t)}_{l,2} \dotplus \ldots \dotplus a^{(t)}_{l,j} \dotplus \hat{y}_1 \dotplus \hat{y}_2 \dotplus \ldots \dotplus \hat{y}_j \tag{18}$$

Let $\Delta D^{(t)}_{k-l} = D^{(t)}_k \dot{-} D^{(t)}_l$, the difference $\Delta d^{(t)}_{k-l,j} = d^{(t)}_{k,j} \dot{-} d^{(t)}_{l,j}$ of the j-th element of the t-th round of diffusion images $D^{(t)}_k$ and $D^{(t)}_l$ in the raster scan order can be obtained, which is

$$\Delta d^{(t)}_{k-l,j} = \left(a^{(t)}_{k,M\times N} \dotplus a^{(t)}_{k,1} \dotplus a^{(t)}_{k,2} \dotplus \ldots \dotplus a^{(t)}_{k,j}\right) \dot{-} \left(a^{(t)}_{l,M\times N} \dotplus a^{(t)}_{l,1} \dotplus a^{(t)}_{l,2} \dotplus \ldots \dotplus a^{(t)}_{l,j}\right). \tag{19}$$

Let

$$\begin{cases} \Delta a^{(t)}_{k-l,M\times N} & = a^{(t)}_{k,M\times N} \dot{-} a^{(t)}_{l,M\times N}, \\ \Delta a^{(t)}_{k-l,1} & = a^{(t)}_{k,1} \dot{-} a^{(t)}_{l,1}, \\ & \vdots \\ \Delta a^{(t)}_{k-l,j-1} & = a^{(t)}_{k,j-1} \dot{-} a^{(t)}_{l,j-1}, \\ \Delta a^{(t)}_{k-l,j} & = a^{(t)}_{k,j} \dot{-} a^{(t)}_{l,j}, \end{cases}$$

and there is

$$\Delta d^{(t)}_{k-l,j} = \Delta a^{(t)}_{k-l,M\times N} \dotplus \Delta a^{(t)}_{k-l,1} \dotplus \Delta a^{(t)}_{k-l,2} \dotplus \ldots \dotplus \Delta a^{(t)}_{k-l,j}. \tag{20}$$

It can be seen from the previous analysis that $P^{(t)} = D^{(t-1)}$, so there is $\Delta P^{(t)}_{k-l} = \Delta D^{(t-1)}_{k-l}$ and $\Delta A^{(t)}_{k-l} = f_{KP}\left(\Delta P^{(t)}_{k-l}\right) = f_{KP}\left(\Delta D^{(t-1)}_{k-l}\right)$, where $f_{KP}(\cdot)$ is the permutation operation on the matrix. Now, let $a^{(t)}_{i,j} = f_{kp_h}\left(p^{(t)}_{i,h}\right)$, where $j = 1,2,\ldots,M\times N, h = 1,2,\ldots,M\times N$, and then, $j = kp_h$ and $h = kp_j^{-1}$ are permutation pairs. $a^{(t)}_{i,j} = f_{kp_h}\left(p^{(t)}_{i,h}\right)$ indicates that the $j = kp_h$-th element $a^{(t)}_{i,j}$ of the i-th permutation image $A^{(t)}_i$ encrypted in the t-th round according to the raster scan order is replaced by the $h = kp_j^{-1}$-th element $p_{i,h}^{(t)}$ of the i-th plaintext image $P_i^{(t)}$ encrypted in the t-th round according to the raster scan order. Thus, the difference of the j-th element between the permutation image $A_k^{(t)}$ and $A_l^{(t)}$ encrypted in the t-th round is

$$\begin{aligned}\Delta a_{k-l,j}^{(t)} &= a_{k,j}^{(t)} \dot{-} a_{l,j}^{(t)} \\ &= f_{kp_h}\left(p_{k,h}^{(t)}\right) \dot{-} f_{kp_h}\left(p_{l,h}^{(t)}\right) \\ &= f_{kp_h}\left(p_{k,h}^{(t)} \dot{-} p_{l,h}^{(t)}\right) \\ &= f_{kp_h}\left(\Delta p_{k-l,h}^{(t)}\right) \\ &= f_{kp_h}\left(\Delta d_{k-l,h}^{(t-1)}\right),\end{aligned} \tag{21}$$

where $\Delta p_{k-l,h}^{(t)} = p_{k,h}^{(t)} \dot{-} p_{l,h}^{(t)}$.

The flow chart for cracking the two-round encryption is shown in Figure 4. The following is a detailed introduction to the two-round encryption cracking algorithm. It should be pointed out that this method is only for the case of two-round encryption with the same permutation matrix.

Step 1: Construct an all-zero plaintext image as P_0 of $M \times N$ for cracking the ciphertext image C of $M \times N$, then construct a plaintext image set $\{P_1, P_2, \ldots, P_{M\times N}\}$, let the k-th element in $P_k (k \in \{1, 2, \ldots, M \times N\})$ according to the raster scan order be 1 and the rest be 0, and this means

$$P_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, P_2 = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \cdots, P_{M\times N} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Step 2: According to Equation (20), the difference relationship between the diffusion images $D_k^{(2)}$ and $D_0^{(2)}$ from their $M \times N$-th to the first element encrypted in the second round is

$$\begin{cases} \Delta d_{k,M\times N}^{(2)} & = \Delta a_{k,M\times N}^{(2)} \dot{+} \Delta a_{k,1}^{(2)} \dot{+} \Delta a_{k,2}^{(2)} \dot{+} \ldots \dot{+} \Delta a_{k,M\times N-2}^{(2)} \dot{+} \Delta a_{k,M\times N-1}^{(2)} \dot{+} \Delta a_{k,M\times N}^{(2)}, \\ \Delta d_{k,M\times N-1}^{(2)} & = \Delta a_{k,M\times N}^{(2)} \dot{+} \Delta a_{k,1}^{(2)} \dot{+} \Delta a_{k,2}^{(2)} \dot{+} \ldots \dot{+} \Delta a_{k,M\times N-2}^{(2)} \dot{+} \Delta a_{k,M\times N-1}^{(2)}, \\ \Delta d_{k,M\times N-2}^{(2)} & = \Delta a_{k,M\times N}^{(2)} \dot{+} \Delta a_{k,1}^{(2)} \dot{+} \Delta a_{k,2}^{(2)} \dot{+} \ldots \dot{+} \Delta a_{k,M\times N-2}^{(2)}, \\ & \vdots \\ \Delta d_{k,2}^{(2)} & = \Delta a_{k,M\times N}^{(2)} \dot{+} \Delta a_{k,1}^{(2)} \dot{+} \Delta a_{k,2}^{(2)}, \\ \Delta d_{k,1}^{(2)} & = \Delta a_{k,M\times N}^{(2)} \dot{+} \Delta a_{k,1}^{(2)}. \end{cases} \tag{22}$$

Modular subtraction of each element from its next adjacent element as

$$\begin{cases} \Delta d_{k,M\times N}^{(2)} \dot{-} \Delta d_{k,M\times N-1}^{(2)} & = \Delta a_{k,M\times N}^{(2)}, \\ \Delta d_{k,M\times N-1}^{(2)} \dot{-} \Delta d_{k,M\times N-2}^{(2)} & = \Delta a_{k,M\times N-1}^{(2)}, \\ \Delta d_{k,M\times N-2}^{(2)} \dot{-} \Delta d_{k,M\times N-3}^{(2)} & = \Delta a_{k,M\times N-2}^{(2)}, \\ & \vdots \\ \Delta d_{k,2}^{(2)} \dot{-} \Delta d_{k,1}^{(2)} & = \Delta a_{k,2}^{(2)}, \\ \Delta d_{k,1}^{(2)} \dot{+} \Delta d_{k,M\times N-1}^{(2)} \dot{-} \Delta d_{k,M\times N}^{(2)} & = \Delta a_{k,1}^{(2)}. \end{cases} \tag{23}$$

Input P_0 and the plaintext image set $P_1, P_2, \ldots, P_{M\times N}$ into the encryption algorithm in turn, and obtain the corresponding ciphertext image C_0 and $C_1, C_2, \ldots, C_{M\times N}$ after two-round encryption, where $D_0^{(2)} = C_0, D_1^{(2)} = C_1, \ldots, D_{M\times N}^{(2)} = C_{M\times N}$. Perform modular subtraction operation on each pixel value in $D_1^{(2)}, D_2^{(2)}, \ldots, D_{M\times N}^{(2)}$ from each pixel value in $D_0^{(2)}$ to obtain $\Delta D_1^{(2)}, \Delta D_2^{(2)}, \Delta D_3^{(2)}, \ldots, \Delta D_{M\times N}^{(2)}$. According to Equation (23), $\Delta A_1^{(2)}, \Delta A_2^{(2)}, \Delta A_3^{(2)}, \ldots, \Delta A_{M\times N}^{(2)}$ are obtained, then the sum of all elements of the above matrices can be calculated as $\sum\limits_{j=1}^{M\times N} \Delta a_{1,j}^{(2)}, \sum\limits_{j=1}^{M\times N} \Delta a_{2,j}^{(2)}, \ldots, \sum\limits_{j=1}^{M\times N} \Delta a_{M\times N,j}^{(2)}$.

Step 3: Because the permutation operation does not change the sum of the element values in the matrix, so $\sum\limits_{j=1}^{M\times N} \Delta d_{1,j}^{(1)} = \sum\limits_{j=1}^{M\times N} \Delta a_{1,j}^{(2)}, \sum\limits_{j=1}^{M\times N} \Delta d_{2,j}^{(1)} = \sum\limits_{j=1}^{M\times N} \Delta a_{2,j}^{(2)}, \ldots, \sum\limits_{j=1}^{M\times N} \Delta d_{M\times N,j}^{(1)} = \sum\limits_{j=1}^{M\times N} \Delta a_{M\times N,j}^{(2)}$.

According to Equation (20), one can obtain

$$\begin{cases} \Delta d_{k,1}^{(1)} & = \Delta a_{k,M\times N}^{(1)} \dotplus \Delta a_{k,1}^{(1)}, \\ \Delta d_{k,2}^{(1)} & = \Delta a_{k,M\times N}^{(1)} \dotplus \Delta a_{k,1}^{(1)} \dotplus \Delta a_{k,2}^{(1)}, \\ & \vdots \\ \Delta d_{k,j}^{(1)} & = \Delta a_{k,M\times N}^{(1)} \dotplus \Delta a_{k,1}^{(1)} \dotplus \Delta a_{k,2}^{(1)} \dotplus \ldots \dotplus \Delta a_{k,j}^{(1)}, \\ & \vdots \\ \Delta d_{k,M\times N}^{(1)} & = \Delta a_{k,M\times N}^{(1)} \dotplus \Delta a_{k,1}^{(1)} \dotplus \Delta a_{k,2}^{(1)} \dotplus \ldots \dotplus \Delta a_{k,M\times N-2}^{(1)} \dotplus \Delta a_{k,M\times N-1}^{(1)} \dotplus \Delta a_{k,M\times N}^{(1)}. \end{cases} \tag{24}$$

From Equation (21), it can be seen that $\Delta a_{k,j}^{(1)} = f_{kp_h}\left(\Delta p_{k,h}^{(1)}\right) = f_{kp_h}(\Delta p_{k,h})$ because $\Delta p_{k,h} = p_{k,h} \dot{-} p_{0,h} = p_{k,h}$. From the properties of the plaintext image P_0 and the constructed plaintext image set $\{P_1, P_2, \ldots, P_{M\times N}\}$, one has

$$\Delta p_{k,h} = \begin{cases} 1 & \text{if } h = k, \\ 0 & \text{if } h \neq k, \end{cases} \tag{25}$$

where $k = 1, 2, \ldots, M \times N$.

Because h and kp_h are a permutation pair, when $h = k$, k and kp_k are a permutation pair. That is, if $p_{k,k}$ is permuted by $a_{k,kp_k}^{(1)}$, $a_{k,kp_k}^{(1)} = p_{k,k} = 1$, then there is

$$\Delta a_{k,j}^{(1)} = \begin{cases} 1 & j = kp_k, \\ 0 & j \neq kp_k, \end{cases}$$

where $kp_k = 1, 2, \ldots, M \times N$, which is substituted into Equation (24), one can obtain

$$kp_k = \begin{cases} 1 & \text{if } \sum\limits_{j=1}^{M\times N} \Delta a_{k,j}^{(2)} = M \times N, \\ M \times N & \text{if } \sum\limits_{j=1}^{M\times N} \Delta a_{k,j}^{(2)} = M \times N + 1, \\ M \times N + 1 - \sum\limits_{j=1}^{M\times N} \Delta a_{k,j}^{(2)} & \text{if } \sum\limits_{j=1}^{M\times N} \Delta a_{k,j}^{(2)} \neq M \times N \text{and} \sum\limits_{j=1}^{M\times N} \Delta a_{k,j}^{(2)} \neq M \times N + 1. \end{cases} \tag{26}$$

According to $\sum\limits_{j=1}^{M\times N} \Delta a_{1,j}^{(2)}, \sum\limits_{j=1}^{M\times N} \Delta a_{2,j}^{(2)}, \ldots, \sum\limits_{j=1}^{M\times N} \Delta a_{M\times N,j}^{(2)}$ obtained in the previous step, the equivalent permutation key KP used for position permutation can be obtained from the above equation.

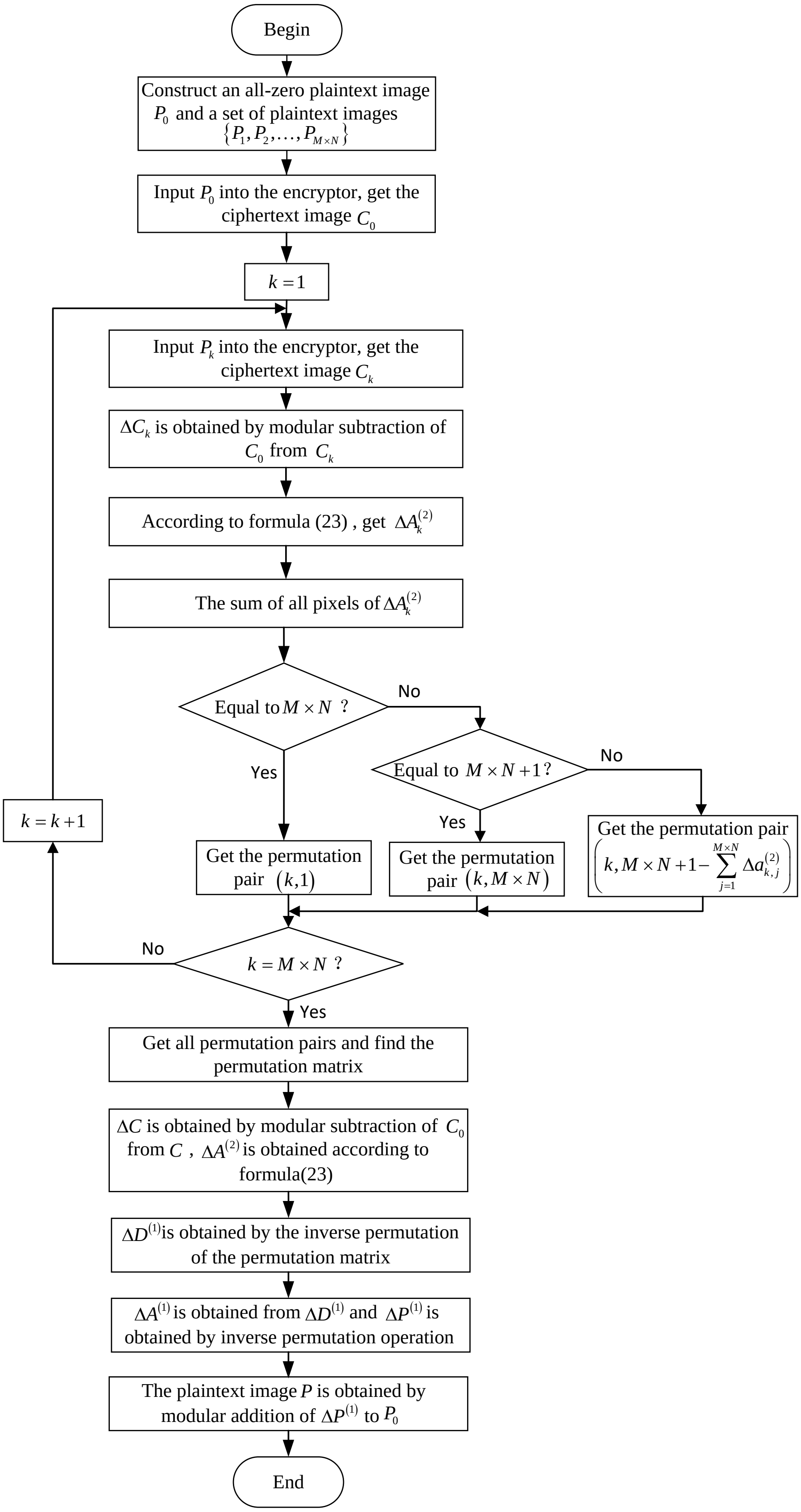

Begin
Construct an all-zero plaintext image P_0 and a set of plaintext images $\{P_1, P_2, \ldots, P_{M\times N}\}$
Input P_0 into the encryptor, get the ciphertext image C_0
$k = 1$
Input P_k into the encryptor, get the ciphertext image C_k
ΔC_k is obtained by modular subtraction of C_0 from C_k
According to formula (23) , get $\Delta A_k^{(2)}$
The sum of all pixels of $\Delta A_k^{(2)}$
Equal to $M \times N$?
No
Yes
Equal to $M \times N + 1$?
No
Yes
$k = k + 1$
Get the permutation pair $(k, 1)$
Get the permutation pair $(k, M \times N)$
Get the permutation pair $\left(k, M \times N + 1 - \sum_{j=1}^{M\times N} \Delta a_{k,j}^{(2)}\right)$
No
$k = M \times N$?
Yes
Get all permutation pairs and find the permutation matrix
ΔC is obtained by modular subtraction of C_0 from C , $\Delta A^{(2)}$ is obtained according to formula(23)
$\Delta D^{(1)}$is obtained by the inverse permutation of the permutation matrix
$\Delta A^{(1)}$ is obtained from $\Delta D^{(1)}$ and $\Delta P^{(1)}$ is obtained by inverse permutation operation
The plaintext image P is obtained by modular addition of $\Delta P^{(1)}$ to P_0
End

Figure 4. The flow chart for cracking the two-round encryption.

Step 4: If the image to be cracked is C, C_0 is the ciphertext image corresponding to the all-zero plaintext. Let $\Delta C = C \dot{-} C_0$, then $\Delta D^{(2)} = \Delta C$. According to Equation (23), the adjacent two elements are modular subtracted to obtain $\Delta A^{(2)}$, and the equivalent permutation key KP is used to obtain $\Delta D^{(1)}$. Similarly, $\Delta A^{(1)}$ and $\Delta P = \Delta P^{(1)}$ can be obtained. Because P_0 is all-zero plaintext, so the deciphered plaintext is $P = P \dot{-} P_0 = \Delta P$.

4.4. Security Analysis of Multi-Round Encryption

The chosen-ciphertext attack method was proposed in [38], which can crack the diffusion operation with ciphertext feedback and different permutation matrices in each round. The applicability of this method was summarized and demonstrated in detail in [40,41]. However, the above literature mainly gave this method for the case without feedback, and then extended it to the case with feedback directly. Through the detailed derivation of the encryption process, the steps of the attack method to crack the case with ciphertext feedback are given in this section. It is not only helpful for understanding the attack method, but also has a good inspiration for guiding the improvement of the algorithm. It should be pointed out that, if there is no special description for multi-round analysis, the symbol definitions given above are still used.

Let $a_{i,j}^{(t)} = f_j\left(p^{(t)}_{i,kp_j^{-1}}\right), j = 1, 2, \ldots, M \times N$; this means that $a_{i,j}^{(t)}$ is the j-th element of the i-th permutation image in raster scan order in the t-th round of encryption is permutated from $p^{(t)}_{i,kp_j^{-1}}$ which is the kp_j^{-1}-th element of the i-th plaintext image $P_i^{(t)}$ in raster scan order in the t-th round of encryption; j and kp_j^{-1} are a permutation pair. The encryption process of the original encryption algorithm can be expressed as a general model as

$$\begin{cases} d_{i,j}^{(t)} & = f_{M\times N}\left(p^{(t)}_{i,kp^{-1}_{M\times N}}\right) \dot{+} \dot{\sum\limits_{u=1}^{j}} f_u\left(p^{(t)}_{i,kp_u^{-1}}\right) \dot{+} \dot{\sum\limits_{u=1}^{j}} \hat{y}_u^{(t)}, \\ p_{i,j}^{(t)} & = d_{i,j}^{(t-1)}, \end{cases} \tag{27}$$

where $i = 0, 1, 2, \cdots$, $j \in \{1, 2, \ldots, M \times N\}$, $t = 1, 2, \ldots, n$, $\dot{+}$ denotes modular addition operation, and $\dot{\sum}$ denotes summation operation of modular addition. As for $u \in \{M \times N, 1, 2, \ldots, j\}$, u and kp_u^{-1} are a permutation pair.

According to Equation (27), one has

$$\begin{aligned} p_{k,j}^{(t)} \dot{-} p_{l,j}^{(t)} &= \Delta p_{k-l,j}^{(t)} = \Delta d_{k-l,j}^{(t-1)} \\ &= f_{M\times N}\left(\Delta p^{(t-1)}_{k-l,kp^{-1}_{M\times N}}\right) \dot{+} \dot{\sum_{u=1}^{j}} f_u\left(\Delta p^{(t-1)}_{k-l,kp_u^{-1}}\right). \end{aligned} \tag{28}$$

According to Equations (21) and (28), $\Delta a_{k-l,r}^{(t)} = a_{k,r}^{(t)} \dot{-} a_{l,r}^{(t)}$ is the difference between the r-th element of the permutation images $A_k^{(t)}$ and $A_l^{(t)}$ in t-th round of encryption. That is,

$$\begin{aligned} \Delta a_{k-l,r}^{(t)} &= f_r\left(\Delta p^{(t)}_{k-l,kp_r^{-1}}\right) \\ &= f_r\left(f_{M\times N}\left(\Delta p^{(t-1)}_{k-l,kp^{-1}_{M\times N}}\right) \dot{+} \dot{\sum_{u=1}^{kp_r^{-1}}} f_u\left(\Delta p^{(t-1)}_{k-l,kp_u^{-1}}\right)\right), \end{aligned} \tag{29}$$

where $k = 1, 2, \cdots$, $l = 0, 1, 2, \cdots$, $r = 1, 2, \ldots, M \times N$, $t = 1, 2, \ldots, n$. By the way, r and kp_r^{-1} are a permutation pair. As for $u \in \{M \times N, 1, 2, \ldots, kp_r^{-1}\}$, u and kp_u^{-1} are a permutation pair.

According to Equation (20), the difference from first to $M \times N$-th element between diffusion image $D_k^{(t)}$ and $D_0^{(t)}$ (that is $l = 0$) in the t-th round of encryption is:

$$\begin{cases} \Delta d_{k,1}^{(t)} &= \Delta a_{k,M\times N}^{(t)} \dot{+} \Delta a_{k,1}^{(t)}, \\ \Delta d_{k,2}^{(t)} &= \Delta a_{k,M\times N}^{(t)} \dot{+} \Delta a_{k,1}^{(t)} \dot{+} \Delta a_{k,2}^{(t)}, \\ & \vdots \\ \Delta d_{k,M\times N}^{(t)} &= \Delta a_{k,M\times N}^{(t)} \dot{+} \dot{\sum\limits_{v=1}^{M\times N}} \Delta a_{k,v}^{(t)}. \end{cases} \tag{30}$$

From Equation (28)–(30), one can further obtain

$$\begin{aligned} \Delta d_{k,1}^{(t)} =& f_{M\times N}\left(f_{M\times N}\left(\Delta p_{k,kp_{M\times N}^{-1}}^{(t-1)}\right) \dot{+} \dot{\sum\limits_{u=1}^{kp_{M\times N}^{-1}}} f_u\left(\Delta p_{k,kp_u^{-1}}^{(t-1)}\right)\right) \\ & \dot{+} f_1\left(f_{M\times N}\left(\Delta p_{k,kp_{M\times N}^{-1}}^{(t-1)}\right) \dot{+} \dot{\sum\limits_{u=1}^{kp_1^{-1}}} f_u\left(\Delta p_{k,kp_u^{-1}}^{(t-1)}\right)\right) \\ \Delta d_{k,2}^{(t)} =& f_{M\times N}\left(f_{M\times N}\left(\Delta p_{k,kp_{M\times N}^{-1}}^{(t-1)}\right) \dot{+} \dot{\sum\limits_{u=1}^{kp_{M\times N}^{-1}}} f_u\left(\Delta p_{k,kp_u^{-1}}^{(t-1)}\right)\right) \\ & \dot{+} f_1\left(f_{M\times N}\left(\Delta p_{k,kp_{M\times N}^{-1}}^{(t-1)}\right) \dot{+} \dot{\sum\limits_{u=1}^{kp_1^{-1}}} f_u\left(\Delta p_{k,kp_u^{-1}}^{(t-1)}\right)\right) \\ & \dot{+} f_2\left(f_{M\times N}\left(\Delta p_{k,kp_{M\times N}^{-1}}^{(t-1)}\right) \dot{+} \dot{\sum\limits_{u=1}^{kp_2^{-1}}} f_u\left(\Delta p_{k,kp_u^{-1}}^{(t-1)}\right)\right) \\ & \vdots \\ \Delta d_{k,M\times N}^{(t)} =& f_{M\times N}\left(f_{M\times N}\left(\Delta p_{k,kp_{M\times N}^{-1}}^{(t-1)}\right) \dot{+} \dot{\sum\limits_{u=1}^{kp_{M\times N}^{-1}}} f_u\left(\Delta p_{k,kp_u^{-1}}^{(t-1)}\right)\right) \\ & \dot{+} \dot{\sum\limits_{v=1}^{M\times N}} f_v\left(f_{M\times N}\left(\Delta p_{k,kp_{M\times N}^{-1}}^{(t-1)}\right) \dot{+} \dot{\sum\limits_{u=1}^{kp_v^{-1}}} f_u\left(\Delta p_{k,kp_u^{-1}}^{(t-1)}\right)\right) \end{aligned}$$

Since the modular addition and permutation can be processed out of order and ended up with the same result, after $t = n$ rounds of encryption, the pixel difference result $\Delta d_{k,j}^{(n)} (j = 1, 2, \ldots, M \times N)$ of diffusion images $D_k^{(n)}$, and $D_0^{(n)}$ can be expressed as the linear combination of the difference pixel point $\Delta p_{k,j}^{(n-1)} (j = 1, 2, \ldots, M \times N)$ of $n - 1$-round plaintext $P_k^{(n-1)}$ and $P_0^{(n-1)}$ modulo 256. $\Delta d_{k,j}^{(n)}, j = 1, 2, \ldots, M \times N$ can be recursively expressed as the linear combination of $\Delta p_{k,j}^{(1)} = \Delta p_{k,j}, j = 1, 2, \ldots, M \times N$ modulo 256, which is

$$\begin{bmatrix} \Delta d_{k,1}^{(n)} \\ \Delta d_{k,2}^{(n)} \\ \vdots \\ \Delta d_{k,M\times N}^{(n)} \end{bmatrix} = \left(\begin{bmatrix} b_{11} & b_{11} & \cdots & b_{1M\times N} \\ b_{21} & b_{22} & \cdots & b_{2M\times N} \\ \vdots & \vdots & \cdots & \vdots \\ b_{M\times N,1} & b_{M\times N,2} & \cdots & b_{M\times N,M\times N} \end{bmatrix} \times \begin{bmatrix} \Delta p_{k,1} \\ \Delta p_{k,2} \\ \vdots \\ \Delta p_{k,M\times N} \end{bmatrix} \right) \mathrm{Mod}256, \tag{31}$$

where Mod represents the modulo of each component of the vector.

For an encryption system, any plaintext image must have only one corresponding ciphertext image. At the same time, any ciphertext image can only be decrypted to one plaintext image; otherwise, the encryption algorithm will not be established. In addition, the number of pixels in the plaintext image and the ciphertext image is constant, so the coefficient matrix is a square matrix, and its rank must be $M \times N$. Furthermore, for the n-round encryption algorithm, the coefficient matrix is represented by the symbol

$$B = \begin{bmatrix} b_{11} & b_{11} & \cdots & b_{1M\times N} \\ b_{21} & b_{22} & \cdots & b_{2M\times N} \\ \vdots & \vdots & \cdots & \vdots \\ b_{M\times N,1} & b_{M\times N,2} & \cdots & b_{M\times N,M\times N} \end{bmatrix}. \tag{32}$$

$\Delta\alpha_k = \left[\Delta d_{k,1}^{(n)}, \Delta d_{k,2}^{(n)}, \cdots, \Delta d_{k,M\times N}^{(n)}\right]^T$ represents the one-dimensional vector converted from the difference matrix between the ciphertext images $D_k^{(n)}$ and $D_0^{(n)}$ in the n-th round of encryption according to the raster scan order. In addition, $\Delta\beta_k = [\Delta p_{k,1}, \Delta p_{k,2}, \cdots, \Delta p_{k,M\times N}]^T$ represents the one-dimensional vector converted from the difference matrix between the plaintext $P_k^{(1)}$ corresponding to the ciphertext image $D_k^{(n)}$ and the plaintext $P_0^{(1)}$ corresponding to the ciphertext image $D_0^{(n)}$, in the n-th round of encryption, according to the raster scan order. Then, the above equation can be expressed as

$$\Delta\alpha_k = (B \times \Delta\beta_k)\text{Mod}256. \tag{33}$$

Consider a set of standard orthogonal bases $e_1, e_2, \ldots, e_{M\times N}$, where $e_1 = [1, 0, \ldots, 0]^T$, $e_2 = [0, 1, 0, \ldots, 0]^T, \ldots, e_{M\times N} = [0, \ldots, 0, 1]^T$. Then, any one-dimensional vector $\Delta\alpha = [c_1, c_2, \ldots, c_{M\times N}]^T$ can be expressed as

$$\Delta\alpha = (c_1 \times e_1 + c_2 \times e_2 + \ldots + c_{M\times N} \times e_{M\times N})\text{Mod}256. \tag{34}$$

According to Equation (33), $e_1, e_2, \ldots, e_{M\times N}$ corresponds to $\Delta\beta_1, \Delta\beta_2, \ldots, \Delta\beta_{M\times N}$, so

$$\begin{cases} e_1 & = (B \times \Delta\beta_1)\text{Mod}256, \\ e_2 & = (B \times \Delta\beta_2)\text{Mod}256, \\ & \vdots \\ e_{M\times N} & = (B \times \Delta\beta_{M\times N})\text{Mod}256, \end{cases} \tag{35}$$

which is substituted into Equation (34) to obtain

$$\begin{aligned} \Delta\alpha =& (c_1 \times (B \times \Delta\beta_1)\text{Mod}256 + c_2 \times (B \times \Delta\beta_2)\text{Mod}256 + \\ & \ldots + c_{M\times N} \times (B \times \Delta\beta_{M\times N})\text{Mod}256)\text{Mod}256 \\ =& (c_1 \times B \times \Delta\beta_1 + c_2 \times B \times \Delta\beta_2 + \ldots + c_{M\times N} \times B \times \Delta\beta_{M\times N})\text{Mod}256 \\ =& (B \times (c_1 \times \Delta\beta_1 + c_2 \times \Delta\beta_2 + \ldots + c_{M\times N} \times \Delta\beta_{M\times N}))\text{Mod}256. \end{aligned} \tag{36}$$

According to Equation (33), $\Delta\beta$ corresponding to $\Delta\alpha$ is

$$\Delta\beta = (c_1 \times \Delta\beta_1 + c_2 \times \Delta\beta_2 + \ldots + c_{M\times N} \times \Delta\beta_{M\times N})\text{Mod}256. \tag{37}$$

The flow chart for cracking the multi-round encryption is shown in Figure 5.

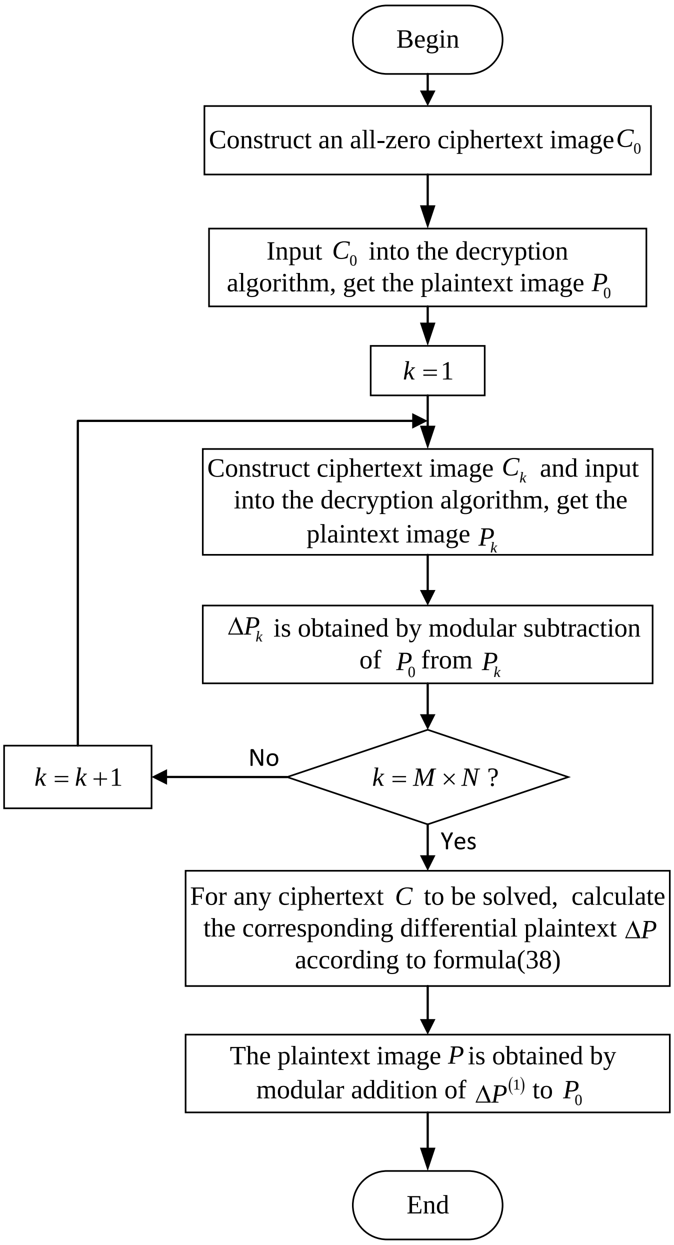

Begin
Construct an all-zero ciphertext image C_0
Input C_0 into the decryption algorithm, get the plaintext image P_0
$k=1$
Construct ciphertext image C_k and input into the decryption algorithm, get the plaintext image P_k
ΔP_k is obtained by modular subtraction of P_0 from P_k
$k = M \times N$?
No
$k = k + 1$
Yes
For any ciphertext C to be solved, calculate the corresponding differential plaintext ΔP according to formula(38)
The plaintext image P is obtained by modular addition of $\Delta P^{(1)}$ to P_0
End

Figure 5. The flow chart for cracking the multi-round encryption.

The details of our cracking process consist of four steps, as given below.

Step 1: Record a ciphertext image of $M \times N$ to be cracked as

$$C = \begin{bmatrix} c_1 & c_2 & \cdots & c_N \\ c_{N+1} & c_{N+2} & \cdots & c_{2\times N} \\ \vdots & \vdots & \cdots & \vdots \\ c_{(M-1)\times N+1} & b_{(M-1)\times N+2} & \cdots & b_{M\times N} \end{bmatrix}.$$

Firstly, an all-zero ciphertext image of $M \times N$ is denoted by C_0, then a ciphertext image set $\{C_1, C_2, \ldots, C_{M\times N}\}$ is constructed, so that the k-th element in C_k according to the raster scan order is set to 1, and the rest is 0 where $k \in \{1,2,\ldots,M \times N\}$, as

$$C_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \cdots, C_{M\times N} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Perform modular subtraction operations of C_0 from $C, C_1, C_2, \ldots, C_{M\times N}$, respectively. Because C_0 is an all-zero ciphertext, the one-dimensional vector converted from $\Delta D^{(n)} = C, \Delta D_1^{(n)} = C_1, \Delta D_2^{(n)} = C_2, \ldots, \Delta D_{M\times N}^{(n)} = C_{M\times N}$ according to the raster scan order is actually the aforementioned $\Delta\alpha, e_1, e_2, \ldots, e_{M\times N}$.

Input $C_0, C_1, C_2, \ldots, C_{M\times N}$ into the decryption machine to obtain a set of corresponding plaintext images, which are denoted as $P_0, P_1, \ldots, P_{M\times N}$.

Step 2: Perform modular subtraction operations of P_0 from $P_1, P_2, \ldots, P_{M\times N}$, respectively, and the result $\Delta P_1 = P_1 \dot{-} P_0, \Delta P_2 = P_2 \dot{-} P_0, \ldots, \Delta P_{M\times N} = P_{M\times N} \dot{-} P_0$ is converted into a one-dimensional vector according to the raster scan order, which is actually the aforementioned $\Delta\beta_1, \Delta\beta_2, \ldots, \Delta\beta_{M\times N}$.

Step 3: Therefore, Equation (37) can also be written as

$$\Delta P = c_1 \times \Delta P_1 \dot{+} c_2 \times \Delta P_2 \dot{+} \ldots \dot{+} c_{M\times N} \times \Delta P_{M\times N}. \tag{38}$$

Step 4: By $\Delta P = \Delta P^{(1)} = P \dot{-} P_0$, the plaintext is finally obtained as

$$P = \Delta P \dot{+} P_0 = (c_1 \times \Delta P_1 \dot{+} c_2 \times \Delta P_2 \dot{+} \ldots \dot{+} c_{M\times N} \times \Delta P_{M\times N}) \dot{+} P_0. \tag{39}$$

5. Numerical Simulation Experiment

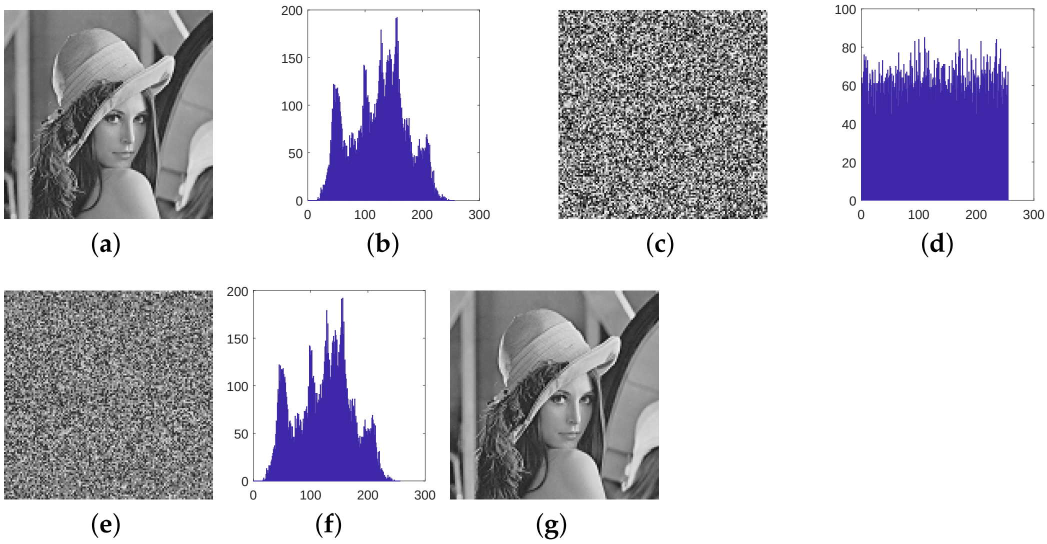
The experimental environment is Intel(R) Core(TM) i5-3230M processor, CPU frequency of 2.60 GHz, 8.00 GB memory, Windows 10 operating system, and MATLABR2021a. The grayscale images Lena, Cameraman, Tiffany, Pepper, Baboon and Sailboat with the size of 128 × 128 are selected for single-round, two-round, and multi-round numerical simulation experiments. The key is selected as follows:

$$\begin{aligned} a_0 &= \{1111010010, 0110101101, 1110001101, 1111110000\}, \\ b_0 &= \{0001111100, 1011100111, 0010010010, 1011000000\}, \\ x &= \{1010101100, 1101011011, 1110001110, 1000100000\}, \\ y &= \{1111111101, 1110100111, 1101011111, 1111001010\}, \\ T &= \{1101000000, 0001011010, 0011001100, 1000011011\}, \\ C_1 = \{1011111100, 0010000101\}, C_2 &= \{1100001010, 1111000110\}. \end{aligned}$$

5.1. Experimental Results

This paper analyzes the original algorithm in the case of single-round, two-round, and multi-round of encryption. Because the first two analysis methods belong to the chosen-plaintext attack and the third one belongs to the chosen-ciphertext attack, the first two analysis methods are stronger in terms of assumptions made and data requirements; they are only applicable to themselves, but the cracking speed is faster. The multi-round analysis method mentioned in this paper is applicable to any number of encryption rounds and has universality.

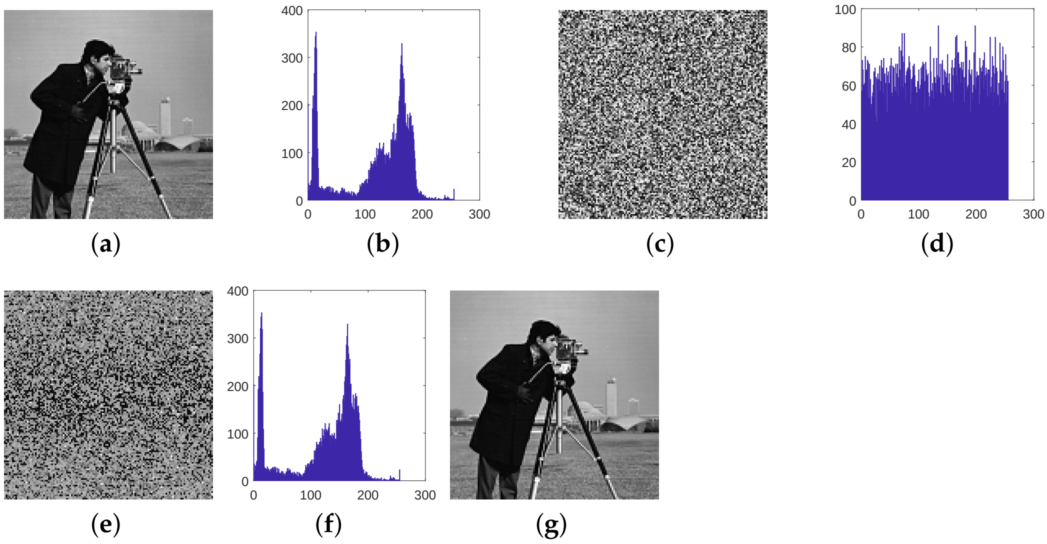
This paper verifies the original encryption algorithm and the analysis results by writing MATLAB programs. According to the original encryption algorithm and the cracking algorithm, the encryption and the decryption programs and cracking programs in single-round, two-round, and multi-round are written, respectively. The related experimental results are shown in Figure 6–12.



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

(**e**) (**f**) (**g**)

Figure 6. The experimental results of cracking the single-round encryption of Lena, (**a**) plaintext image of Lena; (**b**) histogram of plaintext Lena; (**c**) ciphertext of Lena; (**d**) histogram of ciphertext Lena; (**e**) inverse diffusion image of Lena; (**f**) histogram of (**e**); (**g**) the recovered image.



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

(**e**) (**f**) (**g**)

Figure 7. The experimental results of cracking the single-round encryption of Cameraman, (**a**) plaintext image of Cameraman; (**b**) histogram of plaintext Cameraman; (**c**) ciphertext of Cameraman; (**d**) histogram of ciphertext Cameraman; (**e**) inverse diffusion image of Cameraman; (**f**) histogram of (**e**); (**g**) the recovered image.

Figures 6 and 7 respectively list the intermediate experimental results, cracking results and relevant metrics of 128 × 128 Lena image and Cameraman image for single-round encryption. It can be seen that, after encryption, the histogram of ciphertext is uniform and cannot reflect plaintext information. The histogram of intermediate ciphertext obtained by cracking the equivalent diffusion key is the same as that of plaintext, indicating that the diffusion step is cracked. Then, through the obtained permutation equivalent key, the deciphered image is obtained. Compared with the original plaintext image, the deciphered image is exactly the same, which shows that the cracking algorithm is correct.

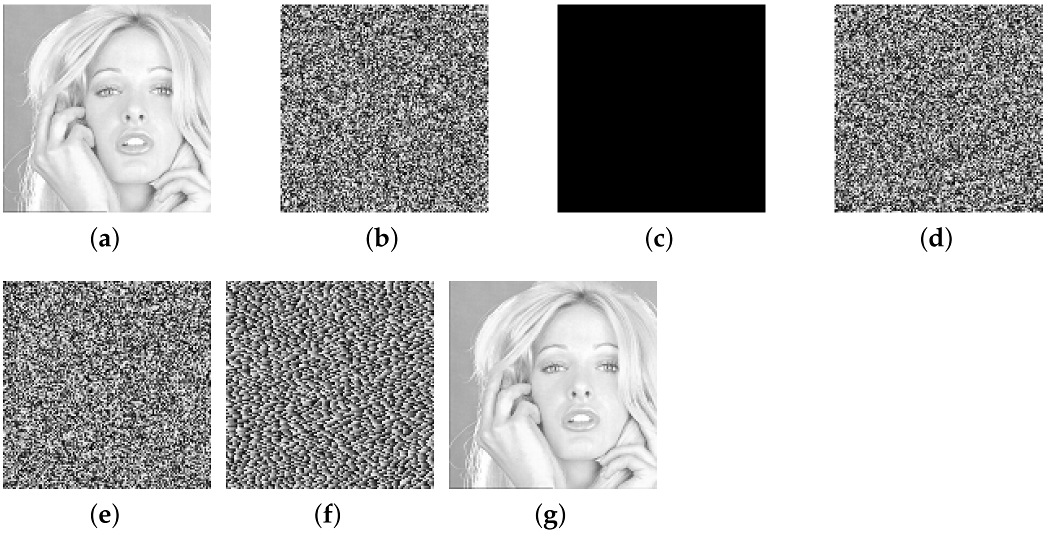


Figure 8. The cracking experiment results of two-round of encryption of Tiffany, (**a**) plaintext image of Tiffany; (**b**) ciphertext of Tiffany; (**c**) all-zero chosen-plaintext P_0; (**d**) the corresponding ciphertext of (**c**); (**e**) calculated differential image ΔC; (**f**) intermediate cracking results $\Delta D^{(1)}$; (**g**) the recovered image.

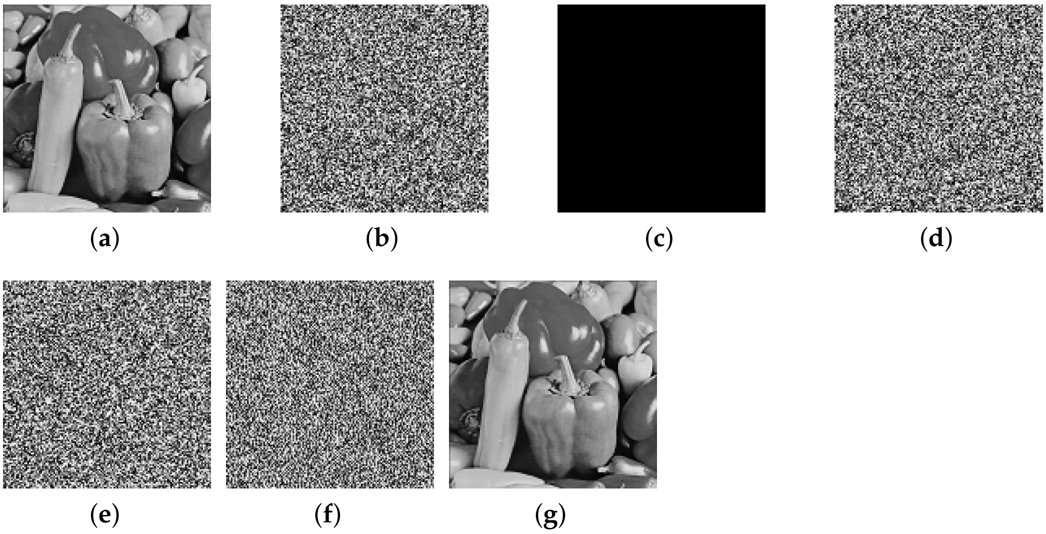


Figure 9. The cracking experiment results of two-round of encryption of Pepper, (**a**) plaintext image of Pepper; (**b**) ciphertext of Pepper; (**c**) all-zero chosen-plaintext P_0; (**d**) the corresponding ciphertext of (**c**); (**e**) calculated differential image ΔC; (**f**) intermediate cracking results $\Delta D^{(1)}$; (**g**) the recovered image.

Figures 8 and 9 respectively list the intermediate experimental results and cracking results of the Tiffany image and Pepper image of 128 × 128 size for two-round of encryption. A set of specially constructed chosen-plaintext is input into the encryption algorithm to obtain a set of plaintext–ciphertext pairs, and according to these plaintext–ciphertext pairs, the permutation matrix is finally cracked. Then, through the differential processing and the obtained permutation matrix, the plaintext is finally cracked, and the comparison between the deciphered image and the plaintext image is exactly the same, which shows that the cracking algorithm is correct.

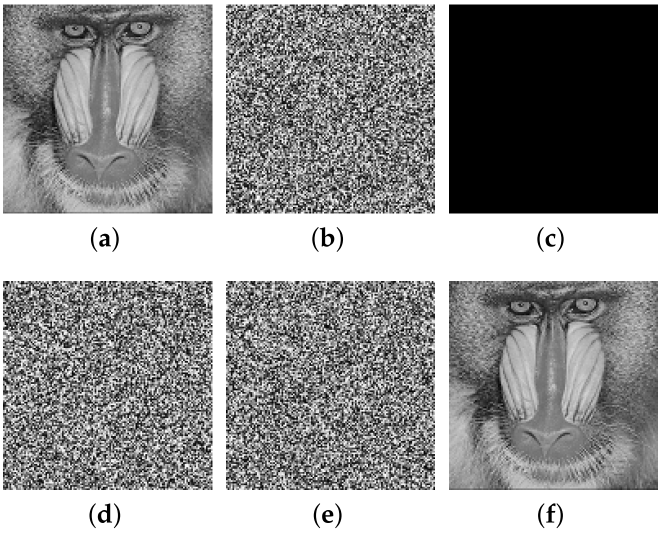


Figure 10. The cracking experiment results of multi-round of encryption of Baboon, (**a**) plaintext image of Baboon; (**b**) ciphertext of Baboon; (**c**) all-zero chosen-ciphertext C_0; (**d**) the corresponding plaintext of (**c**); (**e**) intermediate cracking results ΔP; (**f**) the recovered image.

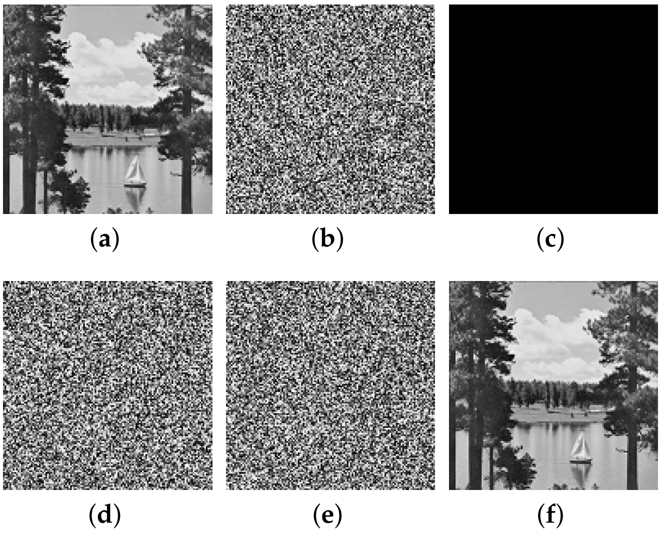


Figure 11. The cracking experiment results of multi-round of encryption of Sailboat, (**a**) plaintext image of Sailboat; (**b**) ciphertext of Sailboat; (**c**) all-zero chosen-ciphertext C_0; (**d**) the corresponding plaintext of (**c**); (**e**) intermediate cracking results ΔP; (**f**) the recovered image.

Figures 10 and 11 list the intermediate experimental results and cracking results of 128 × 128 size Baboon image and Sailboat image for multi-round encryption (without losing generation, the multi-round part is encrypted in three rounds). It can be seen that, by inputting a group of specially constructed chosen-ciphertext into the decryption algorithm, a group of ciphertext–plaintext pairs are obtained, and then the plaintext is finally cracked through differential processing. The comparison between the recovered image and the plaintext image shows that the cracking algorithm is correct.

Theoretically, for a ciphertext image of $M \times N$ size, using multi-round of a cracking algorithm to completely recover the ciphertext image requires the construction of $M \times N$ ciphertext images and an all-zero ciphertext, a total of $M \times N + 1$ ciphertext–plaintext pairs. Because it is difficult to obtain the permission of decryption machine in reality, it is of practical significance to reduce the number of ciphertext–plaintext pairs. Figure 12 shows the effect of constructing different number of ciphertext–plaintext pairs on the finally recovered plaintext image.

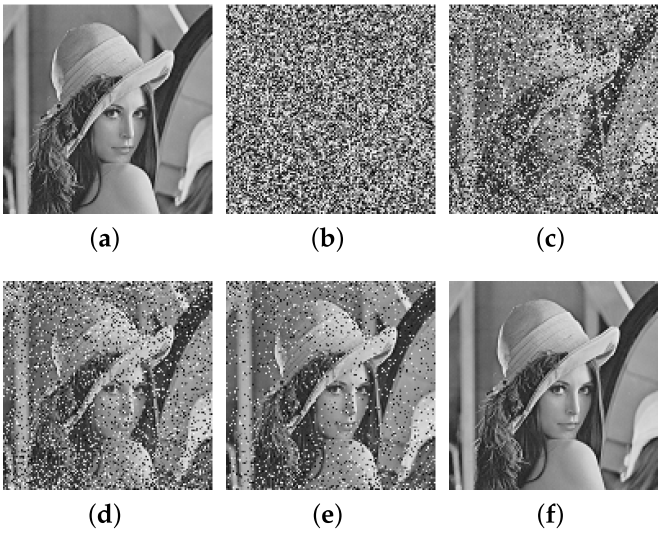

(**a**) (**b**) (**c**)
(**d**) (**e**) (**f**)

Figure 12. The cracking effect of partial ciphertext–plaintext pairs on multi-round cracking algorithm, (**a**) plaintext image of Lena; (**b**) recovered image with 50% ciphertext–plaintext pairs; (**c**) recovered image with 80% ciphertext–plaintext pairs; (**d**) recovered image with 90% ciphertext–plaintext pairs; (**e**) recovered image with 95% ciphertext–plaintext pairs; (**f**) recovered image with 100% ciphertext–plaintext pairs.

It can be seen that the plaintext can be recovered better without some ciphertext–plaintext pairs. In reality, the appropriate number of ciphertext–plaintext pairs can be obtained by constructing an appropriate number of ciphertext, which can reduce the data complexity and improve the cracking efficiency while meeting the cracking requirements.

5.2. Attack Complexity

The attack complexity of cryptanalysis generally includes time complexity and data complexity. However, the time complexity is affected by the performance of the computer and the written cracking program, so it is uncertain. For cryptanalysis, the most important thing is the data complexity, that is, the number of plaintext or ciphertext required to crack an encryption algorithm. The following will discuss the data complexity of the cracking algorithm for the case of single-round, two-round, and multi-round in case of complete cracking.

In the case of single-round, for the grayscale image of $M \times N$, when the key is unknown, the plaintext-ciphertext pair required to decipher the equivalent diffusion key and the equivalent permutation key is $\left(1 + \lceil \log_{256}(M \times N) \rceil\right)$, so the data complexity is $O(\log(M \times N))$.

In the case of two-round, for the grayscale image of $M \times N$, when the key is unknown, the number of chosen-plaintexts required to decipher is $(1 + M \times N)$, so the data complexity is $O(M \times N)$.

In the case of multi-round, for the grayscale image of $M \times N$, when the key is unknown, the number of chosen-ciphertexts required to decipher is $(1 + M \times N)$, so the data complexity is $O(M \times N)$.

5.3. Improvement Plan

From the analysis of this article, it can be seen that the original encryption algorithm has the following vulnerabilities and deficiencies.

(1) The decryption equation of diffusion operation is incorrect. The original decryption result is slightly different from the original encrypted image.

(2) The original encryption algorithm attempts to increase the nonlinear factors by using index matrices and adding a round-down operation to the diffusion equation, for improving the security of the algorithm. However, the analysis found that the above processes can not provide higher security. Instead, an additional permutation is added, resulting in increasing the encryption time. Through the corresponding processing and transformation, the algorithm is still linear and can not resist against the chosen-ciphertext attack.

(3) Under the differential attack, the original diffusion key is completely useless.

In view of the above shortcomings, the following improvement suggestions are put forward.

Cancel the permutation 2 operation. Add new operations in the diffusion process, such as adding S-box, to improve the security of the algorithm. As a nonlinear device, S-box can be applied to the original algorithm to solve its problem. The design of the S-box needs to satisfy cryptographic properties such as nonlinearity, strict avalanche criterion, algebraic immunity, differential uniformity, and correlation immunity [25]. The improvement is given below by taking the S-box as an example.

The permutation operation of the original encryption algorithm is retained. Cancel the permutation operation in the diffusion process of the original encryption algorithm, and complete the diffusion operation according to the original diffusion equation. Take the first 256 bits of the matrix X(if it is less than 256 bits, use the chaotic system to iterate the insufficient bits), obtain a 256-bit index matrix according to the original algorithm, and then form a matrix of 16×16 according to the raster scan method to obtain an S-box. Each pixel of the diffusion image is indexed into the S-box according to the first four bits and the last four bits, and the value of the original pixel is replaced with the corresponding value in the S-box. According to the above steps, the encryption algorithm is carried out for two or more rounds, and the decryption algorithm is the inverse operation of the encryption algorithm.

6. Conclusions

In this paper, the security analysis of the image encryption algorithm based on two-dimensional infinite collapse map is carried out, and the definitions and properties of the round-down operation, the fractional part operation of real numbers, and the modular operation are given. At the same time, by using these theorems, the error of the original diffusion equation is found out, and finally the original encryption algorithm is processed into a general permutation–diffusion structure, and the diffusion structure is processed into a modular addition of the ciphertext feedback and the element of diffusion matrix. On this basis, the single-round, two-round, and multi-round situations are analyzed and discussed respectively. It not only deepens the understanding of the original encryption process, but also helps to guide the improvement of the original encryption algorithm. The correctness of the analysis process is verified by experiments. Finally, the attack complexity is discussed, and suggestions for improvement are given to avoid the shortcomings of the original encryption algorithm.

Author Contributions: Methodology, G.S.; software, G.S.; validation, S.Y. and Q.W.; writing—original draft preparation, G.S.; writing—review and editing, Q.W.; supervision, S.Y. and Q.W.; project administration, S.Y.; funding acquisition, S.Y. and Q.W.

Funding: This research was funded by the Natural Science Foundation of Guangdong Province Grant No. 2022A1515010005 and the National Natural Science Foundation of China under Grants 61901304.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137, https://doi.org/10.1016/j.sigpro.2017.03.011.
2. Oravec, J.; Ovsenik, L.; Papaj, J. An image encryption algorithm using logistic map with plaintext-related parameter values. *Entropy* **2021**, *23*, 1373, https://doi.org/10.3390/e23111373.
3. Parvaz, R.; Zarebnia, M. A combination chaotic system and application in color image encryption. *Opt. Laser Technol.* **2018**, *101*, 30–41, https://doi.org/10.1016/j.optlastec.2017.10.024.
4. Wang, X.; Li, Z. A color image encryption algorithm based on Hopfield chaotic neural network. *Opt. Lasers Eng.* **2019**, *115*, 107–118, https://doi.org/10.1016/j.optlaseng.2018.11.010.
5. Li, T.; Zhang, D. Hyperchaotic image encryption based on multiple bit permutation and diffusion. *Entropy* **2021**, *23*, 510, https://doi.org/10.3390/e23050510.
6. Hua, Z.; Zhu, Z.; Yi, S.; Zhang, Z.; Huang, H. Cross-plane colour image encryption using a two-dimensional logistic tent modular map. *Inf. Sci.* **2021**, *546*, 1063–1083, https://doi.org/10.1016/j.ins.2020.09.032.
7. Pourasad, Y.; Ranjbarzadeh, R.; Mardani, A. A new algorithm for digital image encryption based on chaos theory. *Entropy* **2021**, *23*, 341, https://doi.org/10.3390/e23030341.
8. Naskar, P.K.; Bhattacharyya, S.; Nandy, D.; Chaudhuri, A. A robust image encryption scheme using chaotic tent map and cellular automata. *Nonlinear Dyn.* **2020**, *100*, 2877–2898, https://doi.org/10.1007/s11071-020-05625-3.
9. Roy, S.; Rawat, U.; Sareen, H.A.; Nayak, S.K. IECA: An efficient IoT friendly image encryption technique using programmable cellular automata. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 5083–5102, https://doi.org/10.1007/s12652-020-01813-6.
10. Roy, S.; Shrivastava, M.; Rawat, U.; Pandey, C.V.; Nayak, S.K. IESCA: An efficient image encryption scheme using 2D cellular automata. *J. Inf. Secur. Appl.* **2021**, *61*, 102919, https://doi.org/10.1016/j.jisa.2021.102919.
11. Wang, X.; Guan, N. Chaotic image encryption algorithm based on block theory and reversible mixed cellular automata. *Opt. Laser Technol.* **2020**, *132*, 106501, https://doi.org/10.1016/j.optlastec.2020.106501.
12. Zeng, J.; Wang, C. A novel hyperchaotic image encryption system based on particle swarm optimization algorithm and cellular automata. *Secur. Commun. Netw.* **2021**, *2021*, 6675565, https://doi.org/10.1155/2021/6675565.
13. Zhang, Y. The image encryption algorithm based on chaos and DNA computing. *Multimed. Tools Appl.* **2018**, *77*, 21589–21615, https://doi.org/10.1007/s11042-017-5585-x.
14. Ben Farah, M.A.; Guesmi, R.; Kachouri, A.; Samet, M. A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation. *Opt. Laser Technol.* **2020**, *121*, 105777, https://doi.org/10.1016/j.optlastec.2019.105777.
15. Wu, J.; Liao, X.; Yang, B. Image encryption using 2D Hénon-Sine map and DNA approach. *Signal Process.* **2018**, *153*, 11–23, https://doi.org/10.1016/j.sigpro.2018.06.008.
16. Xu, J.; Zhao, B.; Wu, Z. Research on color image encryption algorithm based on bit-plane and Chen chaotic system. *Entropy* **2022**, *24*, 186, https://doi.org/10.3390/e24020186.
17. Diaconu, A.-V. Circular inter-intra pixels bit-level permutation and chaos-based image encryption. *Inf. Sci.* **2016**, *355–356*, 314–327, https://doi.org/10.1016/j.ins.2015.10.027.
18. Li, Y.; Wang, C.; Chen, H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. *Opt. Lasers Eng.* **2017**, *90*, 238–246, https://doi.org/10.1016/j.optlaseng.2016.10.020.
19. Shahna, K.U.; Mohamed, A. A novel image encryption scheme using both pixel level and bit level permutation with chaotic map. *Appl. Soft Comput.* **2020**, *90*, 106162, https://doi.org/10.1016/j.asoc.2020.106162.
20. Teng, L.; Wang, X.; Meng, J. A chaotic color image encryption using integrated bit-level permutation. *Multimed. Tools Appl.* **2018**, *77*, 6883–6896, https://doi.org/10.1007/s11042-017-4605-1.
21. Xu, L.; Li, Z.; Li, J.; Hua, W. A novel bit-level image encryption algorithm based on chaotic maps. *Opt. Lasers Eng.* **2016**, *78*, 17–25, https://doi.org/10.1016/j.optlaseng.2015.09.007.
22. Singh, L.D.; Singh, K.M. Image encryption using elliptic curve cryptography. *Procedia Comput. Sci.* **2015**, *54*, 472–481, https://doi.org/10.1016/j.procs.2015.06.054.
23. Laiphrakpam, D.S.; Khumanthem, M.S. A robust image encryption scheme based on chaotic system and elliptic curve over finite field. *Multimed. Tools Appl.* **2018**, *77*, 8629–8652, https://doi.org/10.1007/s11042-017-4755-1.
24. Azam, N.A.; Ullah, I.; Hayat, U. A fast and secure public-key image encryption scheme based on Mordell elliptic curves. *Opt. Lasers Eng.* **2021**, *137*, 106371, https://doi.org/10.1016/j.optlaseng.2020.106371.
25. Azam, N.A.; Hayat, U.; Ayub, M. A substitution box generator, its analysis, and applications in image encryption. *Signal Process.* **2021**, *187*, 108144, https://doi.org/10.1016/j.sigpro.2021.108144.
26. Hayat, U.; Azam, N.A. A novel image encryption scheme based on an elliptic curve. *Signal Process.* **2019**, *155*, 391–402, https://doi.org/10.1016/j.sigpro.2018.10.011.
27. Wang, Q.; Yu, S.; Guyeux, C.; Wang, W. Constructing higher-dimensional digital chaotic systems via loop-state contraction algorithm. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 3794–3807, https://doi.org/10.1109/TCSI.2021.3091404.
28. Matthews, R. On the derivation of a “chaotic” encryption algorithm. *Cryptologia* **1989**, *13*, 29–42, https://doi.org/10.1080/0161-118991863745.
29. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284, https://doi.org/10.1142/s021812749800098x.

30. Yu, S.; Lü, J.; Li, C. Some progresses of chaotic cipher and its applications in multimedia secure communications. *J. Electron. Inf. Technol.* **2016**, *38*, 735–752, https://doi.org/10.11999/JEIT151356.
31. Özkaynak, F. Brief review on application of nonlinear dynamics in image encryption. *Nonlinear Dyn.* **2018**, *92*, 305–313, https://doi.org/10.1007/s11071-018-4056-x.
32. Biham, E.; Shamir, A. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptol.* **1991**, *4*, 3–72, https://doi.org/10.1007/bf00630563.
33. Swenson, C. *Modern Cryptanalysis: Techniques for Advanced Code Breaking*; John Wiley & Sons: Indianapolis, IN, USA, 2008; ISBN 978-0-470-13593-8.
34. Solak, E.; Çokal, C.; Yildiz, O.T.; BiyikoǦlu, T. Cryptanalysis of Fridrich's chaotic image encryption. *Int. J. Bifurc. Chaos* **2010**, *20*, 1405–1413, https://doi.org/10.1142/s0218127410026563.
35. Xie, E.Y.; Li, C.; Yu, S.; Lü, J. On the cryptanalysis of Fridrich's chaotic image encryption scheme. *Signal Process.* **2017**, *132*, 150–154, https://doi.org/10.1016/j.sigpro.2016.10.002.
36. Fu, C.; Meng, W.; Zhan, Y.; Zhu, Z.; Lau, F.C.M.; Tse, C.K.; Ma, H. An efficient and secure medical image protection scheme based on chaotic maps. *Comput. Biol. Med.* **2013**, *43*, 1000–1010, https://doi.org/10.1016/j.compbiomed.2013.05.005.
37. Chen, L.; Wang, S. Differential cryptanalysis of a medical image cryptosystem with multiple rounds. *Comput. Biol. Med.* **2015**, *65*, 69–75, https://doi.org/10.1016/j.compbiomed.2015.07.024.
38. Chen, L.; Ma, B.; Zhao, X.; Wang, S. Differential cryptanalysis of a novel image encryption algorithm based on chaos and Line map. *Nonlinear Dyn.* **2016**, *87*, 1797–1807, https://doi.org/10.1007/s11071-016-3153-y.
39. Hu, Y. Research on the Cryptanalysis of a Class of Image Chaotic Cipher Using Permutation–Diffusion Approach. Doctoral Dissertation, Guangdong University of Technology, Guangzhou, China, 2021.
40. Chen, J.; Chen, L.; Zhou, Y. Cryptanalysis of image ciphers with permutation-substitution network and chaos. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 2494–2508, https://doi.org/10.1109/TCSVT.2020.3021908.
41. Chen, J.; Chen, L.; Zhou, Y. Universal chosen-ciphertext attack for a family of image encryption schemes. *IEEE Trans. Multimed.* **2021**, *23*, 2372–2385, https://doi.org/10.1109/TMM.2020.3011315.
42. Cao, W.; Mao, Y.; Zhou, Y. Designing a 2D infinite collapse map for image encryption. *Signal Process.* **2020**, *171*, 107457, https://doi.org/10.1016/j.sigpro.2020.107457.
43. Rosen, K.H. *Elementary Number Theory and Its Applications*, 5th ed.; China Machine Press: Beijing, China, 2005; pp. 7–8, ISBN 7-111-15914-4.
44. Johnsonbaugh, R. *Discrete Mathematics*, 7th ed.; Publishing House of Electronics Industry: Beijing, China, 2009; p. 120, ISBN 978-7-121-08534-5.
45. Li, S.; Li, C.; Chen, G.; Bourbakis, N.G.; Lo, K.T. A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Process. Image Commun.* **2008**, *23*, 212–223, https://doi.org/10.1016/j.image.2008.01.003.