


Article

# Improved Cryptanalysis and Enhancements of an Image Encryption Scheme Using Combined 1D Chaotic Maps

Congxu Zhu <sup>1,2,3,\*</sup> , Guojun Wang <sup>2</sup> and Kehui Sun <sup>4</sup>

<sup>1</sup> School of Information Science and Engineering, Central South University, Changsha 410083, China

<sup>2</sup> School of Computer Science and Technology, Guangzhou University, Guangzhou 510006, China; csgjwang@163.com

<sup>3</sup> School of Computer and Science, Liaocheng University, Liaocheng 252059, China

<sup>4</sup> School of Physics and Electronics, Central South University, Changsha 410083, China; kehui@csu.edu.cn

\* Correspondence: zhucx@csu.edu.cn; Tel.: +86-0731-8882-7601

Received: 4 October 2018; Accepted: 31 October 2018; Published: 3 November 2018



**Abstract:** This paper presents an improved cryptanalysis of a chaos-based image encryption scheme, which integrated permutation, diffusion, and linear transformation process. It was found that the equivalent key streams and all the unknown parameters of the cryptosystem can be recovered by our chosen-plaintext attack algorithm. Both a theoretical analysis and an experimental validation are given in detail. Based on the analysis of the defects in the original cryptosystem, an improved color image encryption scheme was further developed. By using an image content-related approach in generating diffusion arrays and the process of interweaving diffusion and confusion, the security of the cryptosystem was enhanced. The experimental results and security analysis demonstrate the security superiority of the improved cryptosystem.

**Keywords:** image encryption; chaotic cryptography; cryptanalysis; chosen-plaintext attack; image information entropy

## 1. Introduction

The transmission of a digital image from the public network is becoming more and more frequent nowadays. Consequently, it is urgent to guarantee the security and privacy of image transmission, especially for military images and some sensitive content images. As an essential technical means, image encryption approaches are particularly important in image communications. However, traditional cryptography cannot quickly encrypt images with large amounts of data. As traditional cryptography relies on the complexity of computation, it is not easy to generate a large number of keys quickly. In this application background, chaotic encryption is a good complement to traditional cryptography, especially in image encryption. As chaotic signals have some excellent characteristics required by cryptography, chaotic systems have become a fine tool for information encryption [1], especially for image encryption applications. Due to this, chaotic systems have been widely used in designing image encryption algorithms. Entropy is an important measure of the chaotic characteristics of dynamical systems. Entropy, chaos and information theory are closely related [2–4].

Among many chaos-based algorithms for encrypting an image, the permutation and diffusion (PD) structure encryption algorithm, proposed by Fridrich [5], has become a typical model. This structure consists of a permutation (i.e., pixel position scrambling) procedure and a diffusion (i.e., pixel value alteration) procedure. Based on a typical model, researchers have tried many different ways of improving innovation. Some studies have proposed different image permutation strategies [6–12]. Some researchers have proposed novel image diffusion techniques [10,13–17]. Many researchers have

attempted to improve the performance of image encryption systems through other improvements. References [18–24] improve the performance of secret key streams through a new chaotic system model. References [25–29] improve the anti-attack performance of a cryptographic algorithm by introducing a plaintext-related mechanism in generating the key streams. References [30–36] introduce the DNA coding principle in bioinformatics to enhance the security of the algorithms. References [37–41] focus on improving the speed of image encryption algorithms through comprehensive means. In References [42–44], the S-boxes are applied to the design of efficient image encryption algorithm and combined with transformation technology, the performance of the image encryption algorithm is improved. In References [45–47], wavelet analysis technology is introduced into the field of image encryption and the ideas are novel. In References [48,49], fractal analysis technology is investigated, which is related to chaos and has a good application potential in image encryption.

Another research direction closely related to encryption is cryptanalysis. The goal of cryptanalysis is to find a way to decipher secret keys or plaintext without knowing the secret keys of encryption systems [50–53]. Cryptanalysis can also find out flaws in encryption algorithms and can help cryptographic system designers to improve the security performance of cryptographic algorithms, which can avoid losses caused by potential vulnerabilities and make valuable contributions to encryption. Hence, cryptanalysis can also promote cryptography. Recent cryptanalysis research shows that some chaos-based image encryption algorithms have some security flaws and the cryptosystems can be broken by using various attack methods. For example, we launched a chosen-plaintext attack [54] on the scheme in Reference [55]. Wu [53] broke the encryption scheme in Reference [56] by a chosen-plaintext attack.

Except for the security performance, efficiency is another important issue of an image encryption scheme for practical applications. For this reason, image ciphers with a higher speed are consequently more desirable than those with a low speed. It is well known that low-dimensional discrete chaotic systems need less time than high-dimensional continuous-time chaotic systems to generate chaotic sequences of the same length. Therefore, using a low dimensional discrete chaotic system as a key generator of a cryptosystem has a higher speed. Furthermore, the complexity of discrete chaotic systems is much larger than those of the continuous-time chaotic systems [57–59] and the cipher image encrypted with a chaotic sequence of much larger complexity has a higher security. Therefore, using a 1D discrete chaotic system to encrypt color images, not only has the advantage of fast speed but also has the advantage of higher security. In Reference [60], Pak proposed a color image encryption scheme (denoted as Pak's cryptosystem hereinafter) by using combined 1D chaotic maps. Pak's system has the merits of a simple structure, high speed and a relatively high safety. It is a pity that Pak's algorithm cannot resist the chosen-plaintext attack. To the best of our knowledge, so far, only Wang [61] and Chen [62] have done cryptanalysis on Pak's scheme. Unfortunately, neither of the two previous analyses can crack all of the unknown parameters of Pak's encryption scheme due to the difficulty of the comprehensive cryptanalysis. Therefore, the previous cryptanalysis is incomplete. Moreover, Wang's cryptanalysis scheme has obvious problems and a very low efficiency, while Chen's cryptanalysis scheme did not give the specific process of deciphering the permutation secret keys. In order to overcome the shortcomings of the above cryptanalysis work, this paper presents a more comprehensive and efficient cryptanalysis on Pak's cryptosystem. With our improved cryptanalysis, both equivalent secret keys and all of the unknown parameters of Pak's cryptosystem can be completely deciphered.

Despite its security flaws, Pak's encryption scheme still has many advantages to carry forward. Its design idea is clear and novel, and its efficiency is relatively high. Therefore, it is worth preserving these advantages and improving their defects. For this reason, this paper further proposes an improved enhanced color image encryption scheme, which includes both an image content-related approach in generating diffusion arrays and the process of interweaving diffusion and confusion.

The rest of this paper is organized as follows. Section 2 describes briefly Pak's algorithm and the related cryptanalysis. The improved cryptanalysis and attacks on Pak's algorithm are presented in Section 3. An enhanced encryption scheme is proposed in Section 4. Some experimental results and

analysis for the enhanced scheme are given in Section 5. Finally, some concluding remarks are given in Section 6.

## 2. Description of Pak’s Scheme and the Related Cryptanalysis

Pak’s algorithm includes three processing stages. (1) Confusion: Pixel level permutation; (2) Diffusion: Pixel values encryption; (3) Linear transformation. Before the encrypting process, the 3D RGB color image with  $M$  row  $N$  columns is converted into a 1D pixel array  $\mathbf{P} = [p(1), p(2), \dots, p(L)]$  according to the R, G, and B components successively, where  $L = M \times N \times 3$ . Each value of  $p(i)$  is an integer in the range  $[0, 255]$ . The flow of Pak’s encryption scheme can be visualized in Figure 1. Where,  $\mathbf{P}$  is the plain image pixel array, and  $\mathbf{C}'$  is the final cipher image pixel array. SSS represents the combined Sine-Sine chaotic System.  $\mathbf{X}'$  is the permutation position array and  $\mathbf{D}'$  is the diffusion array. Both  $\mathbf{X}'$  and  $\mathbf{D}'$  are generated by chaotic sequences.

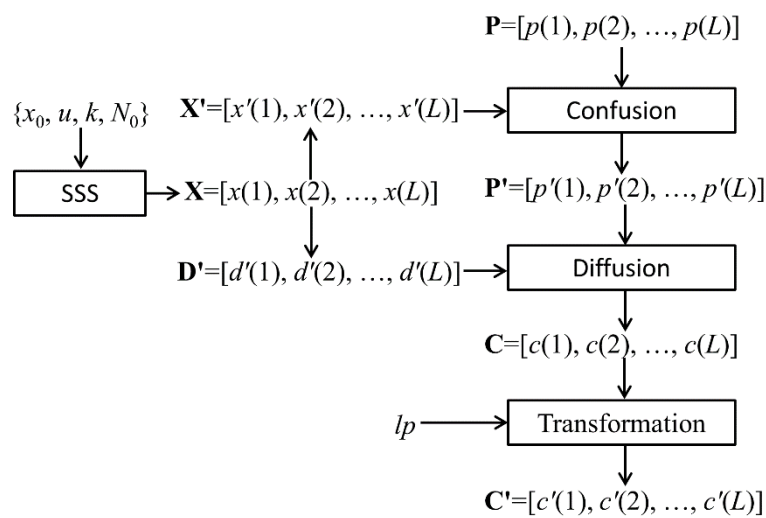


Figure 1. A flow chart of Pak’s encryption scheme.

### 2.1. The New Chaotic System

The chaotic system adopted in Pak’s encryption scheme is a newly discovered chaotic map by using the chaotic sine map, which is expressed as

$$x(n + 1) = u \times \sin[\pi \times x(n)] \times 2^k - \lfloor u \times \sin[\pi \times x(n)] \times 2^k \rfloor \tag{1}$$

where,  $u$  is the control parameter of the system and  $\{x(n), n = 0, 1, 2, \dots\}$  is the output chaotic sequence with the initial value  $x(0) = x_0$ .  $\lfloor x \rfloor$  is the largest integer that is smaller than or equal to  $x$ . System (1) is called a Sine-Sine system (SSS) [60], which is chaotic when  $u \in (0, 10]$  and  $k \in [8, 20]$ . Parameters  $k, u$  and  $x_0$  were used as secret keys.

### 2.2. The Confusion Process

In the confusion process, a permutation operation is performed on the pixel level with a position transformation. The operational process consists of the following steps:

Step 1: By using specified parameter values  $x_0, u$  and  $k$ , iterate the new chaotic system  $(N_0 + L)$  times and select the rear  $L$  elements to make a sub chaotic sequence  $\mathbf{X} = [x(1), x(2), \dots, x(L)]$ . Where  $N_0$  is an integer used as a security key.

Step 2: Sequence  $\mathbf{X}$  is sorted in ascending order. Then, one can obtain a sorted chaotic sequence  $\mathbf{SX} = [sx(1), sx(2), \dots, sx(L)]$  and a permutation position array  $\mathbf{X}' = [x'(1), x'(2), \dots, x'(L)]$ , where  $x'(i)$  are integers ranging from 1 to  $L$ . If  $x(i) = sx(j)$ , then  $x'(i) = j$ .

Step 3: Get the permuted image pixel sequence  $\mathbf{P}' = [p'(1), p'(2), \dots, p'(L)]$  by using the permutation position array  $\mathbf{X}'$  and the plain image pixel sequence  $\mathbf{P}$ . The transformation relation is

$$p'(i) = p(x'(i)). \tag{2}$$

### 2.3. Diffusion Process

In the diffusion process, pixel value encryption is performed based on a diffusion array  $\mathbf{D}'$ . The operational process consists of the following two steps: The operational process consists of the following steps:

Step 1: Generate the diffusion array  $\mathbf{D}' = [d'(1), d'(2), \dots, d'(L)]$  from the chaotic sequence  $\mathbf{X}$  as:

$$d'(i) = \text{mod}(\lfloor x(i) \times 10^k \rfloor, 256). \tag{3}$$

Step 2: Get the temporary ciphered image pixel array  $\mathbf{C} = [c(1), c(2), \dots, c(L)]$  from the diffusion vector  $\mathbf{D}'$  and the permuted image array  $\mathbf{P}'$  according to the following diffusion equation:

$$\begin{cases} c(i) = \text{mod}(p'(i) + d'(i), 256) \oplus \text{seed}, \text{ if } i = 1, \\ c(i) = \text{mod}(p'(i) + d'(i), 256) \oplus c(i - 1), \text{ if } i > 1, \end{cases} \tag{4}$$

where  $\oplus$  denotes the binary XOR operator.  $c(i - 1)$  is the previous cipher pixel, and *seed* is a preset constant.

### 2.4. Linear Transformation

Get the final cipher image pixel array  $\mathbf{C}' = [c'(1), c'(2), \dots, c'(L)]$  from the temporary cipher image pixel array  $\mathbf{C}$  and a security number *lp* as

$$\begin{cases} c'(i - lp) = c(i), \text{ if } i > lp, \\ c'(i - lp + L) = c(i), \text{ if } i \leq lp, \end{cases} \tag{5}$$

where *lp* is used as a security key. In order to see the result of the linear transformation at a glance, we used a graph to express the linear transformation process, which is shown in Figure 2. There are two key points in this linear transformation process, which deserve our special attention. One, the first pixel in the array  $\mathbf{C}$  was moved to the  $(L - lp + 1)$  position in the array  $\mathbf{C}'$ , that is  $c'(L - lp + 1) = c(1)$ . Second, the original two adjacent pixels  $c(lp)$  and  $c(lp + 1)$  were moved to the end and start of the array  $\mathbf{C}'$ , that is,  $c'(L) = c(lp)$ ,  $c'(1) = c(lp + 1)$ . If  $lp = 0$  or  $lp = L$ , then  $c'(i) = c(i)$ . Hence, a reasonable range of *lp* is  $0 < lp < L$ .

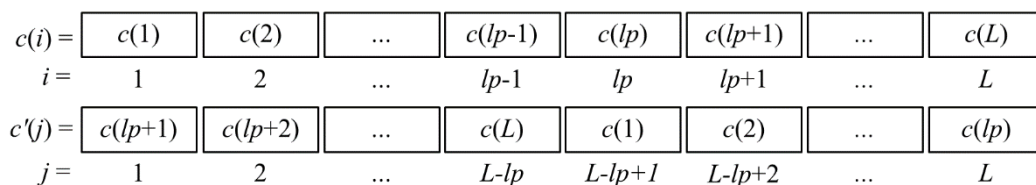


Figure 2. The linear transformation operation.

The final cipher image was obtained by converting the 1D pixel vector  $\mathbf{C}'$  into a 2D color image consisting of R, G and B components with the size of  $M \times N$ . The secret keys used in Pak's algorithm consists of five parameters  $\{x_0, u, k, N_0, lp\}$ .

The decryption process is the inverse operation of the encryption process and it was omitted here.

According to Kerchhoff's principle, when analyzing an encryption algorithm, an assumption is made that the cryptanalyst knows exactly the design and working of the cryptosystem. Namely, the only thing the attacker does not know is the secret key. The definition of a chosen-plaintext attack can be described

as follows: Attackers have the chance to use the encryption machine temporarily, hence they can select a special plaintext to encrypt and get its corresponding ciphertext without knowing the secret keys.

In Pak's algorithm, the permutation position array  $\mathbf{X}'$  and the diffusion array  $\mathbf{D}'$  are determined by parameters  $\{x_0, u, k, N_0\}$  and have nothing to do with the plain image. Namely,  $\mathbf{X}'$  and  $\mathbf{D}'$  are static and do not change with different images to be encrypted. The secret key,  $lp$ , and the unknown parameter  $seed$  also have nothing to do with the plain image. Therefore, attackers can choose some special plaintext images to encrypt by using Pak's encryption machine when they temporarily obtain the opportunity to use Pak's encryption machine and obtain the corresponding ciphertext image to use these known plaintext-ciphertext image pairs to crack the equivalent key sequences  $\mathbf{X}'$ ,  $\mathbf{D}'$ , parameters  $lp$  and  $seed$ . By using these equivalent key sequences  $\mathbf{X}'$  and  $\mathbf{D}'$ , parameters  $lp$  and  $seed$ , any image encrypted by Pak's encryption machine can be decrypted without knowing the original keys of Pak's encryption machine. This is the basic principle of the chosen-plaintext attack model. According to this attack model, it is obvious that Pak's scheme cannot resist a chosen-plaintext attack.

### 2.5. The Related Cryptanalysis Work

In Wang's cryptanalysis scheme, the authors constructed an equivalent cryptosystem for Pak's cryptosystem. In the equivalent cryptosystem, they constructed the new permutation position array  $\mathbf{X}''$  and diffusion array  $\mathbf{D}''$  of the equivalent encryption scheme by transforming the original permutation position array  $\mathbf{X}'$  and diffusion array  $\mathbf{D}'$  with the secret parameter  $lp$  respectively. The relationships of the key streams between the equivalent cryptosystem and Pak's cryptosystem are as follows

$$\begin{cases} x''(i - lp) = x'(i), & \text{if } i \in (lp, L], \\ x''(i - lp + L) = x'(i), & \text{if } i \in [1, lp]. \end{cases} \quad (6)$$

$$\begin{cases} d''(i - lp) = d'(i), & \text{if } i \in (lp, L], \\ d''(i - lp + L) = d'(i), & \text{if } i \in [1, lp]. \end{cases} \quad (7)$$

Wang's equivalent encryption scheme contains only two processes: permutation and diffusion, which can be described by Equations (8) and (9) respectively.

$$p''(i) = p(x''(i)) \quad (8)$$

$$c''(i) = \text{mod}(p''(i) + d''(i), 256) \oplus c''(i - 1) \quad (9)$$

where  $\mathbf{P}'' = [p''(1), p''(2), \dots, p''(L)]$  is the permuted image pixel sequence of Wang's equivalent cryptosystem, which has the following relations with  $\mathbf{P}'$  in Pak's system

$$\begin{cases} p''(i - lp) = p'(i), & \text{if } i \in (lp, L], \\ p''(i - lp + L) = p'(i), & \text{if } i \in [1, lp]. \end{cases} \quad (10)$$

$\mathbf{C}'' = [c''(1), c''(2), \dots, c''(L)]$  is the final cipher image pixel array of Wang's equivalent cryptosystem. The authors claim that  $c''(i) = c'(i)$  will hold if the Equations (6)–(10) hold.

The operation process of Wang's chosen-plaintext attack scheme is divided into the following three stages.

(1) Extract the diffusion array  $\mathbf{D}''$ . Select a special plain-image  $\mathbf{P}$  consisting of all 0 elements such that  $p''(i) = 0$  and obtain the corresponding cipher-image  $\mathbf{C}''$ . According to Equation (9), the diffusion array  $\mathbf{D}''$  is extracted as

$$d''(i) = c''(i) \oplus c''(i - 1). \quad (11)$$

(2) Extract the permutation position array  $\mathbf{X}''$ . Select  $L$  special plain images with the 1D pixel arrays respectively denoted as  $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_L$  and the  $j$ th element in the pixel array  $\mathbf{P}_j$  is 1; all other elements are 0. Get the corresponding encrypted image arrays  $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_L$ . By using one plain

image  $\mathbf{P}_j$  and the corresponding  $\mathbf{C}_j$ , only one element  $x''(i)$  in  $\mathbf{X}''$  can be obtained. All elements of  $\{x''(1), x''(1), \dots, x''(L)\}$  can be obtained when  $L$  pairs of  $(\mathbf{P}_j, \mathbf{C}_j)$  are used.

(3) Recover the original plain image. By using the new permutation position array  $\mathbf{X}''$  and the new diffusion array  $\mathbf{D}''$ , recover the original plain image  $\mathbf{P}$  from the target cipher image  $\mathbf{C}$ .

We find that Wang's cryptanalysis algorithm has the following issues:

(1) The authors assume that the attacker knows the parameter *seed* and used it as a known parameter in the equivalent encryption system. In fact, the *seed* parameter is a constant set in Pak's cryptosystem. Although the attacker can use Pak's encryption machine temporarily, the *seed* parameter is unknown to the attacker.

(2) Although the authors claim that the cipher image  $\mathbf{C}''$  obtained from their equivalent cryptosystem is the same as the cipher image  $\mathbf{C}'$  obtained by the original Pak's cryptosystem, no strict proof is given. In fact, the cipher image pixel array  $\mathbf{C}''$  is not equivalent to  $\mathbf{C}'$  due to the unknown parameter  $lp$ , which is not broken out by the authors. The proof procedure is as follows.

When encrypting the first pixel by Wang's equivalent cryptosystem, Equation (9) is degenerated into the form as  $c''(1) = \text{mod}(p''(1) + d''(1), 256) \oplus c''(0)$ , where  $c''(0)$  is not a pixel value of the array  $\mathbf{C}''$  and  $c''(0)$  may be the parameter *seed*. From Equations (7), (9) and (10), we can get  $p''(1) = p'(lp + 1)$  and  $d''(1) = d'(lp + 1)$ . Then one can obtain  $c''(1)$  as

$$c''(1) = \text{mod}(p'(lp + 1) + d'(lp + 1), 256) \oplus \text{seed}. \quad (12)$$

while  $c'(1)$  obtained by using Pak's algorithm is as

$$c'(1) = \text{mod}(p'(lp + 1) + d'(lp + 1), 256) \oplus c'(L). \quad (13)$$

By comparing Equations (12) with (13),  $c''(1) \neq c'(1)$ .

When  $i = L - lp + 1$ , Equation (9) is degenerated into the form  $c''(L - lp + 1) = \text{mod}(p''(L - lp + 1) + d''(L - lp + 1), 256) \oplus c''(L - lp)$ . From Equations (7), (9) and (10), we can get  $p''(L - lp + 1) = p'(1)$  and  $d''(L - lp + 1) = d'(1)$ . As a result,  $c''(L - lp + 1)$  is as

$$c''(L - lp + 1) = \text{mod}(p'(1) + d'(1), 256) \oplus c''(L - lp). \quad (14)$$

while using Pak's algorithm,  $c'(L - lp + 1)$  is as

$$c'(L - lp + 1) = \text{mod}(p'(1) + d'(1), 256) \oplus \text{seed}. \quad (15)$$

Comparing Equations (14) with (15),  $c''(L - lp + 1) \neq c'(L - lp + 1)$ .

Based on  $c''(1) \neq c'(1)$ , one can deduce that  $c''(i) \neq c'(i)$ ,  $i = 2, 3, \dots, L$ .

In fact, there are some defects in Wang's cryptanalysis algorithm because the authors completely ignore the role of the parameter  $lp$  and do not break out  $lp$ . However, when the parameter  $lp$  is not known, one cannot know where the *seed* should be used to calculate  $c''(i)$ .

(3) The most serious problem in Wang's cryptanalysis scheme is that the number of chosen plain images is too high to reach  $M \times N \times 3$  in extracting the permutation position array  $\mathbf{X}''$ . The use of one chosen plain image at a time can only break one element value of  $\mathbf{X}''$ , which is very inefficient, so Wang's cryptanalysis scheme is unrealistic.

In Chen's cryptanalysis scheme, unfortunately, the parameter *seed* is also not deciphered and used as a known parameter. Thus, reducing the difficulty of the cryptanalysis. In addition, Chen did not give the specific process of deciphering the permutation position array  $\mathbf{X}'$ .

### 3. The Improved Cryptanalysis Scheme

In order to provide a more comprehensive and efficient cryptanalysis method on Pak's encryption algorithm, we propose an improved chosen-plaintext attack algorithm to Pak's scheme. Suppose the

target color cipher image to be decrypted has the size of  $L = M \times N \times 3$ . Firstly, we cracked the secret parameter  $lp$  and the diffusion array  $[d'(2), d'(3), \dots, d'(L)]$  except for  $d'(1)$  by using two selected plain images and their corresponding cipher images. Secondly, we cracked the unknown parameter  $seed$  and  $d'(1)$  by using one or more than one selected plain images. Thirdly, we cracked the permutation position array  $\mathbf{X}'$  by using  $\lceil (M \times N \times 3)/255 \rceil$  selected plain images and their corresponding cipher images, where  $\lceil x \rceil$  is the smallest integer that is greater than or equal to  $x$ . Wang's cryptanalysis algorithm needs  $M \times N \times 3$  selected plain images to decipher the permutation position array  $\mathbf{X}'$ , while our cryptanalysis algorithm only needs  $\lceil (M \times N \times 3)/255 \rceil$  selected plain images to decipher the permutation position array  $\mathbf{X}'$ . Hence, the efficiency of our improved chosen-plaintext attack algorithm is about 255 times that of Wang's algorithm.

### 3.1. Recover the Secret Key $lp$ and the Diffusion Array

According to Equations (4) and (5),  $d'(i)$  can be calculated as

$$\begin{cases} d'(1) = c'(L - lp + 1) \oplus seed - p'(1), \text{ if } i = 1, \\ d'(i) = c'(L - lp + i) \oplus c'(L - lp + i - 1) - p'(i), \text{ if } 1 < i \leq lp, \\ d'(lp + 1) = c'(1) \oplus c'(L) - p'(lp + 1), \text{ if } i = lp + 1, \\ d'(i) = c'(i - lp) \oplus c'(i - lp - 1) - p'(i), \text{ if } lp + 1 < i \leq L, \end{cases} \quad (16)$$

where  $x - y = \text{mod}(x - y + 256, 256)$ . Obviously, if the  $seed$  in Equation (16) is replaced by  $c'(L - lp)$ , then the relationship between  $d'(i)$  and  $c'(j)$  can be expressed in Figure 3.

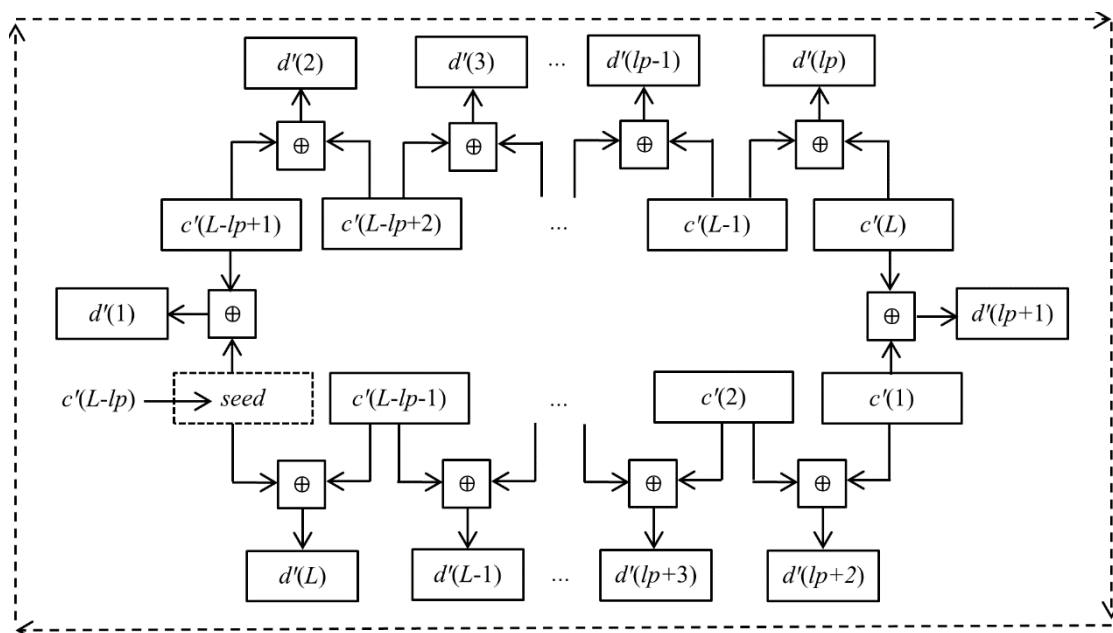


Figure 3. The diagram of the relationship between  $\mathbf{D}'$  and  $\mathbf{C}'$ .

From Figure 3, one can see that each key  $d'(i)$  is related to a pair of adjacent pixel values  $\{c'(j), c'(j + 1)\}$  or  $\{c'(L), c'(1)\}$ . To avoid the influence of the unknown parameter  $lp$ , we can select a specific plain image where all pixels  $p'(i)$  have the same value  $q$ , then we can calculate a series of values by neighbors  $\{c'(L), c'(1)\}, \{c'(1), c'(2)\}, \{c'(2), c'(3)\}, \dots, \{c'(L - 1), c'(L)\}$  and store these values in a temporary array  $\mathbf{D} = [d(1), d(2), \dots, d(L)]$ , where  $d(i)$  is as

$$\begin{cases} d(1) = c'(1) \oplus c'(L) - q, \\ d(i) = c'(i) \oplus c'(i - 1) - q, \quad i = 2, 3, \dots, L. \end{cases} \quad (17)$$

Equation (17) brings us great convenience for computing  $d(i)$  because it does not contain the unknown parameters  $lp$  and  $seed$ . Obviously, the equivalent relationship of the elements between  $\mathbf{D}' = [d'(1), d'(2), \dots, d'(L)]$  and  $\mathbf{D} = [d(1), d(2), \dots, d(L)]$  is as follows

$$\begin{cases} d'(i) = d(i - lp), \text{ if } i > lp, \\ d'(i) = d(L + i - lp), \text{ if } i \leq lp. \end{cases} \quad (18)$$

Namely,  $d'(lp + 1) = d(1)$ ,  $d'(lp + 2) = d(2)$ ,  $\dots$ ,  $d'(L - 1) = d(L - lp - 1)$ ,  $d'(L) = d(L - lp)$ ;  $d'(1) = d(L - lp + 1)$ ,  $d'(2) = d(L - lp + 2)$ ,  $\dots$ ,  $d'(lp - 1) = d(L - 1)$ ,  $d'(lp) = d(L)$ .

It is worth noting that, except for  $d'(1)$  or  $d(L - lp + 1)$ , the rest of the values  $d(i)$  ( $i \neq L - lp + 1$ ) obtained by Equation (17) are all right values. Namely, when calculating  $d(L - lp + 1)$ , if we do not use the parameter  $seed$  and use the  $c'(L - lp)$  value instead of  $seed$ , then the result of  $d(L - lp + 1)$  may be wrong. Considering the values of  $d(i)$  or  $d'(i)$  are determined by parameters  $\{x_0, u, k, N_0\}$  and have nothing to do with the content of the image, if we choose two different plain images and get the corresponding cipher images, by using the two pairs of plaintext-ciphertext to calculate  $d_1(i)$  and  $d_2(i)$ , then one can find the only position of  $ii$  that the value of  $d_1(ii)$  and  $d_2(ii)$  will not be identical but values of  $d_1(i)$  and  $d_2(i)$  at other locations  $i$  ( $i \neq ii$ ) are definitely the same. Once the location  $ii$  is sought out, the value of  $lp$  can be determined, which is  $lp = L + 1 - ii$ .

Based on the above idea, we get the algorithm for deciphering the secret key parameter  $lp$  and the diffusion array  $d'(i)$ , which is described as follows:

Step 1: Let  $q = 0$ , and select a special plain image  $\mathbf{PA} = [pa(1), pa(2), \dots, pa(L)]$  that all pixels  $pa(i)$  have the same value  $q$  and obtain the corresponding cipher image  $\mathbf{CA}' = [ca'(1), ca'(2), \dots, ca'(L)]$  by using Pak's encryption machinery. As  $\mathbf{PA}' = [q, q, \dots, q]$ , then we can get a array  $\mathbf{DA} = [da(1), da(2), \dots, da(L)]$  by using Equation (17).

Step 2: Let  $q = q + 1$ , and select a special plain image  $\mathbf{PB} = [pb(1), pb(2), \dots, pb(L)]$  that all pixels  $pb(i)$  have the same value  $q$ . Obtain the corresponding cipher-image  $\mathbf{CB}' = [cb'(1), cb'(2), \dots, cb'(L)]$  by using Pak's encryption machine. Because  $\mathbf{PB}' = [q, q, \dots, q]$ , then we can get another array  $\mathbf{DB} = [db(1), db(2), \dots, db(L)]$  by using Equation (17).

Step 3: Compare  $da(i)$  and  $db(i)$  one by one for  $i = 1, 2, \dots, L$ . If it exists at position  $I = ii$  and meets the relationship  $da(ii) \neq db(ii)$ , then  $L - lp + 1 = ii$ , so  $lp$  is determined as  $lp = L + 1 - ii$ , and go to Step 4. Otherwise, repeat Step 2 to Step 3 until  $lp$  is determined.

Step 4: After the value of  $lp$  is ascertained, we can recover the diffusion array  $\mathbf{D}'$  of Pak's cryptosystem by using Equation (18). Where only the value of  $d'(1)$  is incorrect.

### 3.2. Recover $d'(1)$ and the Unknown Parameter Seed

According to the first formula in Equation (16),  $(d'(1), seed)$  meets the following relationship

$$c'(L - lp + 1) = \text{mod}(d'(1) + p'(1), 256) \oplus seed. \quad (19)$$

Using the special chosen plain image  $\mathbf{PA} = [0, 0, \dots, 0]$  and  $\mathbf{PB} = [1, 1, \dots, 1]$ , we have got a pair of ciphertext data  $(ca'(L - lp + 1), cb'(L - lp + 1))$  in the previous section. Therefore,  $d'(1)$  and  $seed$  needs to satisfy the following equation:

$$\begin{cases} ca'(L - lp + 1) = \text{mod}(d'(1) + 0, 256) \oplus seed, \\ cb'(L - lp + 1) = \text{mod}(d'(1) + 1, 256) \oplus seed. \end{cases} \quad (20)$$

Consider such a fact that  $seed \in \{0, 1, 2, \dots, 255\}$  and  $d'(1) \in \{0, 1, 2, \dots, 255\}$ , so the solution of Equation (20) can be easily obtained by the computer exhaustive algorithm. However, the solution



$[d'(1), seed]$  of Equation (20) is not unique because the equations in Equation (20) are not two linear equations. Suppose an equation for  $d'$  and  $seed$  has the following form:

$$\text{mod}(d' + q, 256) \oplus seed = c'. \tag{21}$$

Regarding the solutions of Equation (21), We have the following Proposition:

**Proposition 1.** For any values of  $q \in Z_{256}$  and  $c' \in Z_{256}$ , if  $[d', seed]$  is a solution of Equation (21), then  $[\text{mod}(d' + 128, 256), \text{mod}(seed + 128, 256)]$  is also a solution of Equation (21). Where  $d' \in Z_{256}$  and  $seed \in Z_{256}$ .

**Proof.** Suppose the binary value of  $\text{mod}(d' + q, 256)$  is  $(d_8d_7d_6d_5d_4d_3d_2d_1)_2$  and the binary value of  $seed$  is  $(s_8s_7s_6s_5s_4s_3s_2s_1)_2$ .

If  $d_8 = 0$ , then  $\text{mod}(\text{mod}(d' + 128, 256) + q, 256) = \text{mod}(d' + q + 128, 256) = \text{mod}(\text{mod}(d' + q, 256) + 128, 256) = \text{mod}((0d_7d_6d_5d_4d_3d_2d_1)_2 + (10000000)_2, 256) = (1d_7d_6d_5d_4d_3d_2d_1)_2 = (\bar{d}_8d_7d_6d_5d_4d_3d_2d_1)_2$ . Where,  $\bar{x}$  represents the binary inverse value of  $x$ .

If  $d_8 = 1$ , then  $\text{mod}(\text{mod}(d' + 128, 256) + q, 256) = \text{mod}(d' + q + 128, 256) = \text{mod}(\text{mod}(d' + q, 256) + 128, 256) = \text{mod}((1d_7d_6d_5d_4d_3d_2d_1)_2 + (10000000)_2, 256) = (0d_7d_6d_5d_4d_3d_2d_1)_2 = (\bar{d}_8d_7d_6d_5d_4d_3d_2d_1)_2$ .

If  $s_8 = 0$ , then  $\text{mod}(seed + 128, 256) = \text{mod}((0s_7s_6s_5s_4s_3s_2s_1)_2 + (10000000)_2, 256) = (1s_7s_6s_5s_4s_3s_2s_1)_2 = (\bar{s}_8s_7s_6s_5s_4s_3s_2s_1)_2$ .

If  $s_8 = 1$ , then  $\text{mod}(seed + 128, 256) = \text{mod}((1s_7s_6s_5s_4s_3s_2s_1)_2 + (10000000)_2, 256) = (0s_7s_6s_5s_4s_3s_2s_1)_2 = (\bar{s}_8s_7s_6s_5s_4s_3s_2s_1)_2$ .  $\square$

Considering  $\bar{d}_8 \oplus \bar{s}_8 = d_8 \oplus s_8$ , we can obtain that  $\text{mod}(\text{mod}(d' + 128, 256) + q, 256) \oplus \text{mod}(seed + 128, 256) = (\bar{d}_8d_7d_6d_5d_4d_3d_2d_1)_2 \oplus (\bar{s}_8s_7s_6s_5s_4s_3s_2s_1)_2 = (d_8d_7d_6d_5d_4d_3d_2d_1)_2 \oplus (s_8s_7s_6s_5s_4s_3s_2s_1)_2 = c'$ . This means that  $[\text{mod}(d' + 128, 256), \text{mod}(seed + 128, 256)]$  is also a solution of Equation (21).

Suppose Equation (20) has  $m$  groups of solutions ( $m \geq 2$ ) as  $[d'_1(1), seed_1], [d'_2(1), seed_2], \dots, [d'_m(1), seed_m]$ . If  $m = 2$ , then the two groups of solutions are all the required results and the task of recovering  $(d'(1), seed)$  has been completed. If  $m > 2$ , then we must select some other plain image  $\mathbf{P} = [q, q, \dots, q]$  and obtain the corresponding cipher image  $\mathbf{C}' = [c'(1), c'(2), \dots, c'(L)]$ , where  $q > 1$ . In addition, we can obtain another equation as:  $\text{mod}(d'(1) + q, 256) \oplus seed = c'(L - lp + 1)$ . Under the constraint of the other equation, we can remove those superfluous solutions that do not satisfy all equations until the remaining solutions are only 2 groups. In this way, the unknown parameter  $seed$  and the secret key  $d'(1)$  of the original encryption system can be deciphered. The concrete algorithm for recovering  $d'(1)$  and  $seed$  is described as follows:

Step 1: Let  $m$  groups of solutions of Equation (20) be saved in the array  $\mathbf{R} = [r(1), r(2), \dots, r(m)]$  and  $\mathbf{S} = [s(1), s(2), \dots, s(m)]$  sequentially, Where  $r(i) = d'_i(1), s(i) = seed_i, i = 1, 2, \dots, m$ . Let  $q = 1$ .

Step 2: Check the value of  $m$ . If  $m \leq 2$ , then go to Step 9. If  $m > 2$ , then go to Step 3.

Step 3:  $q = q + 1$ .

Step 4: For  $i = 1, 2, \dots, m$ , each groups of solutions  $[r(i), s(i)]$  is assumed to be used to encrypt the plaintext pixel value  $q$  and calculate the corresponding ciphertext values as  $cc(i) = \text{mod}(q + r(i), 256) \oplus s(i)$ .

Step 5: For  $i = 1, 2, \dots, m$ , Check whether the value of each element in the array  $[cc(1), cc(2), \dots, cc(m)]$  is exactly the same. If  $cc(i)$  is exactly the same, then repeat Step 3 to Step 5. If  $cc(i)$  is not exactly the same, then go to Step 6.

Step 6: Select a special plain image array  $\mathbf{P} = [q, q, \dots, q]$  and obtain the corresponding cipher image pixels array  $\mathbf{C}' = [c'(1), c'(2), \dots, c'(L)]$  by using Pak's encryption machine.

Step 7: For each solution group  $[r(i), s(i)]$ , calculate the values of  $\text{mod}(r(i) + q, 256) \oplus s(i), i = 1, 2, \dots, m$ . If  $\text{mod}(r(i) + q, 256) \oplus s(i) \neq c'(L - lp + 1)$ , then delete the  $i$ -th solution group  $[r(i), s(i)]$  from  $\mathbf{S}$  and  $\mathbf{R}$  respectively.

Step 8: Modify the value of  $m$ , that is,  $m = \text{size}(\mathbf{R})$ , and return to Step 2.

Step 9: Output the final values of  $[d'(1), seed]$ , that is  $[d'(1), seed] = [r(1), s(1)]$  or  $[d'(1), seed] = [r(2), s(2)]$ .

### 3.3. Recover the Permutation Position Array $X'$

After the RGB image matrix is converted into a 1D gray image pixel sequence  $\mathbf{P} = [p(1), p(2), \dots, p(L)]$ , array  $\mathbf{P}$  has  $L$  pixels and  $L = M \times N \times 3$ . Each value of  $p(i)$  is an integer in the range of  $[0, 255]$ . If  $L \leq 255$ , then only one chosen-plain image  $\mathbf{P} = [1, 2, \dots, L]$  is necessary to recover the permutation position array  $X'$ , so that each pixel in the chosen plain image has different values in  $\{1, 2, \dots, L\}$ . If  $L > 255$ , then  $n$  chosen plain images are required to recover the permutation position array  $X'$ , where  $n = \lceil L/255 \rceil > 1$ . In this case, we select a series of special color plain images  $(\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n)$  and  $\mathbf{P}_j = [p_j(1), p_j(2), \dots, p_j(L)]$ . We divide  $\mathbf{P}_j$  into  $n$  groups and each group contains 255 pixels except for the last one and the last group contains  $q$  pixels ( $q \leq 255$ ). For the  $j$ -th chosen-plain image pixel array  $\mathbf{P}_j$ , we assign each element of the  $j$ -th group a distinct value between 1 to 255 and the others are assigned the value of 0. The patterns of elements in each chosen plain image pixel array  $\mathbf{P}_j$  are shown in Figure 4.

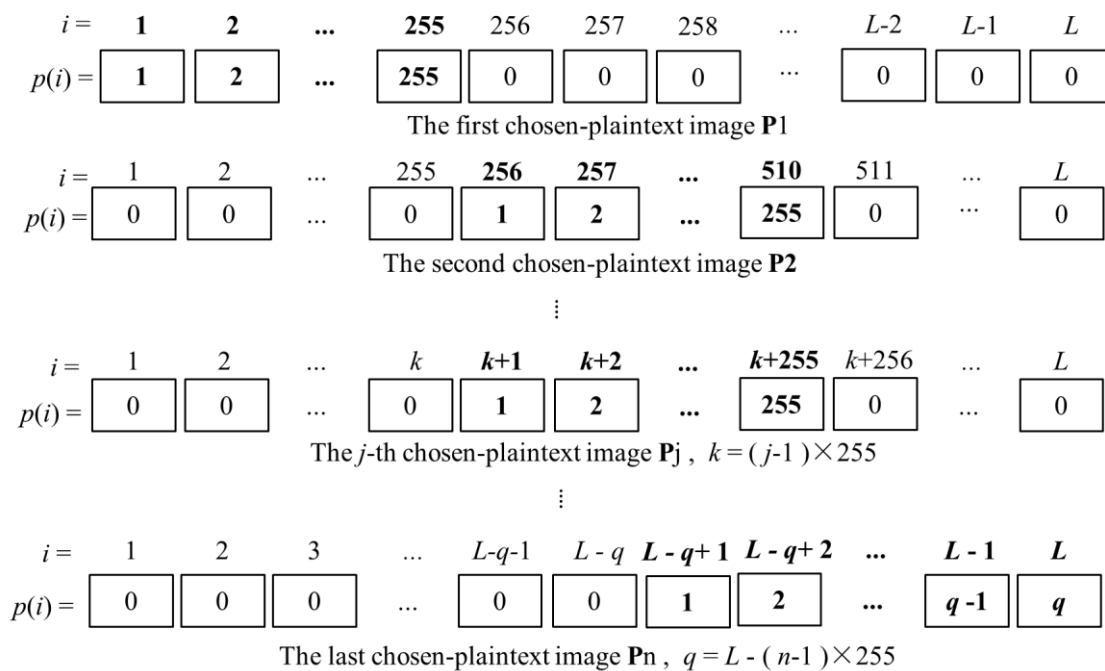


Figure 4. The patterns of elements in each chosen plain image pixel array  $\mathbf{P}_j$ .

We then obtain the corresponding series of cipher images  $(\mathbf{C}'_1, \mathbf{C}'_2, \dots, \mathbf{C}'_n)$  by using Pak's encryption machine. Where,  $\mathbf{C}'_j = [c'_j(1), c'_j(2), \dots, c'_j(L)]$ . Then, we can decrypt  $\mathbf{C}'_j$  to obtain  $\mathbf{P}'_j = [p'_j(1), p'_j(2), \dots, p'_j(L)]$ , where  $p'_j(i)$  can be obtained by using Equation (16).

Finally, because of the relationship  $p'_j(i) = p_j(x'(i))$  ( $i \in [1, L]$ ),  $X'$  can be determined by comparing  $\mathbf{P}'_j$  and  $\mathbf{P}_j$ . Namely, if  $p'_j(i) = p_j(k)$ , then  $x'(i) = k$ .

### 3.4. Recover the Original Plain Image

In Section 3.1 to 3.3, we obtained the secret keys  $\{lp, X', D'\}$  and the unknown parameter *seed*, which are unrelated to the plain image or ciphertext image. Therefore, we can decrypt any other ciphertext image  $\mathbf{CI}$  by using the parameter set  $\{seed, lp, X', D'\}$ . The decryption process to recover the plain image  $\mathbf{PI}$  from the target ciphertext image  $\mathbf{CI}$  is exactly the same as the decryption process of Pak's scheme, which can be described as follows:

Step 1: Convert the color ciphertext image  $\mathbf{CI}$  with a size of  $M \times N \times 3$  into a 1D pixel array  $\mathbf{C}' = [c'(1), c'(2), \dots, c'(L)]$ , where  $L = M \times N \times 3$ .

Step 2: Obtain the intermediary cipher image pixel array  $\mathbf{C} = [c(1), c(2), \dots, c(L)]$  from the final cipher pixel array  $\mathbf{C}' = [c'(1), c'(2), \dots, c'(L)]$  by performing the inverse transformation of Equation (5).

Step 3: Recover the permuted image pixel array  $\mathbf{P}' = [p'(1), p'(2), \dots, p'(L)]$  by performing the inverse diffusion process of Equation (4).

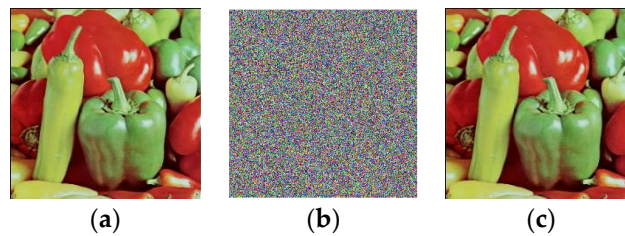
Step 4: Do inverse permutation on  $\mathbf{P}'$  to obtain  $\mathbf{P}$  by using the inverse permutation process of Equation (2).

Step 5: Convert the 1D array  $\mathbf{P}$  into a 3D matrix with a size of  $M \times N \times 3$  and the original color plain image  $\mathbf{PI}$  is recovered.

### 3.5. Examples of the Improved Cryptanalysis Scheme

Suppose the right values of original secret keys in Pak's cryptosystem are as follows:  $x_0 = 0.456$ ,  $u = 5.4321$ ,  $k = 14$ ,  $N_0 = 1000$ ,  $lp = 5$ , and  $seed = 250$ .

**Example 1.** In this example, the plain image  $\mathbf{P}$  is the color peppers with a size of  $256 \times 256 \times 3$ . The plain image and its cipher image encrypted by using Pak's encryption machine are shown in Figure 5a,b respectively. The deciphered image by using our chosen-plaintext attack is shown in Figure 5c, which is exactly the same as the original plain image in Figure 5a. Through the image peppers as an example, our attack attains demonstration.



**Figure 5.** The experimental results of the chosen-plaintext attacks. (a) The plain image; (b) the cipher image; (c) the cracked image.

**Example 2.** The secret key parameters are the same as those of Example 1. In order to verify the correctness of our chosen-plaintext attack scheme more intuitively, this example shows a simple and specific numerical experiment. In this example, the plain image  $\mathbf{PI}$  is the color image with size of  $2 \times 2 \times 3$  ( $L = M \times N \times 3 = 12$ ), and its components are as

$$\mathbf{P}_R = \begin{bmatrix} 11 & 13 \\ 12 & 14 \end{bmatrix}, \mathbf{P}_G = \begin{bmatrix} 21 & 23 \\ 22 & 24 \end{bmatrix}, \mathbf{P}_B = \begin{bmatrix} 31 & 33 \\ 32 & 34 \end{bmatrix}. \tag{22}$$

Its corresponding 1D pixel array  $\mathbf{P}$  is:

$$\mathbf{P} = [11, 12, 13, 14, 21, 22, 23, 24, 31, 32, 33, 34]. \tag{23}$$

As the result, the 1D pixel array  $\mathbf{C}'$  encrypted by Pak's encryption machine is:

$$\mathbf{C}' = [246, 16, 1, 6, 37, 3, 137, 197, 162, 215, 51, 22]. \tag{24}$$

By choosing two special plain-image array  $\mathbf{PA} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$  and  $\mathbf{PB} = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$ , we obtain the corresponding cipher image arrays as  $\mathbf{CA}' = [173, 117, 137, 108, 97, 110, 17, 229, 170, 195, 20, 18]$  and  $\mathbf{CB}' = [255, 38, 219, 61, 51, 35, 163, 218, 138, 224, 56, 63]$ . According to Equation (16), we obtain  $\mathbf{DA}$  and  $\mathbf{DB}$  as:  $\mathbf{DA} = [191, 216, 252, 229, 13, 15, 127, \mathbf{244}, 79, 105, 215, 6]$ ,  $\mathbf{DB} = [191, 216, 252, 229, 13, 15, 127, \mathbf{120}, 79, 105, 215, 6]$ . By comparing  $\mathbf{DA}$  and  $\mathbf{DB}$ , we find that  $da(8) \neq db(8)$ , then  $ii = 8$ , and  $lp = L + 1 - ii = 12 + 1 - 8 = 5$ . Then, Equation (19) is changed into the following form

$$\begin{cases} ca'(8) = 229 = \text{mod}(d'(1) + 0, 256) \oplus seed \\ cb'(8) = 218 = \text{mod}(d'(1) + 1, 256) \oplus seed \end{cases}$$

which has four groups of solution:  $[d'(1), seed] = \{[31, 250], [95, 186], [159, 122], [223, 58]\}$ . For  $q = 2, 3, \dots$ , check the values of “ $\text{mod}(d'(1) + q, 256) \oplus seed$ ” with the four groups of solution. When  $q = 33$ , we find that “ $\text{mod}(d'(1) + q, 256) \oplus seed$ ” has different values (186, 58, 186, 58) corresponding to the four groups of solution. We then select a special color plain image  $\mathbf{P} = [33, 33, \dots, 33]$  and obtain the corresponding cipher image  $\mathbf{C}' = [127, 134, 155, 157, 179, 131, 35, 186, 202, 64, 184, 159]$  by using Pak's encryption machine, in which  $c'(8) = 186$ . Then we can determine that (31, 250) and (159, 122) are two right groups of secret keys to  $[d'(1), seed]$ . If we adopt  $[d'(1), seed] = [159, 122]$  as the secret keys, then we can obtain  $\mathbf{D}'$  from  $\mathbf{DA}$  or  $\mathbf{DB}$  by using Equation (18), that is,  $\mathbf{D}' = [159, 79, 105, 215, 6, 191, 216, 252, 229, 13, 15, 127]$ .

To recover the permutation position array  $\mathbf{X}'$ , we select a special color plain image  $\mathbf{P} = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$  and obtain the corresponding cipher image  $\mathbf{C}' = [240, 44, 45, 220, 207, 217, 89, 211, 132, 239, 53, 58]$  by using Pak's encryption machine. Then we can obtain its intermediary ciphertext array  $\mathbf{C}$  according to Equation (5) as  $\mathbf{C} = [211, 132, 239, 53, 58, 240, 44, 45, 220, 207, 217, 89]$ . Then we can obtain the permuted pixel array  $\mathbf{P}'$  from  $\mathbf{D}'$  by using Equation (4), that is,  $\mathbf{P}' = [10, 8, 2, 3, 9, 11, 4, 5, 12, 6, 7, 1]$ . By comparing  $\mathbf{P}$  and  $\mathbf{P}'$ , the permutation array  $\mathbf{X}'$  is recovered as  $\mathbf{X}' = [10, 8, 2, 3, 9, 11, 4, 5, 12, 6, 7, 1]$ .

For the target ciphertext array  $\mathbf{C}'$  of Equation (24), we obtain its intermediary cipher pixel array  $\mathbf{C}$  according to Equation (5) as  $\mathbf{C} = [197, 162, 215, 51, 22, 246, 16, 1, 6, 37, 3, 137]$ . Then we obtain the permuted pixel array  $\mathbf{P}'$  from  $\mathbf{D}'$  by using Equation (4), that is,  $\mathbf{P}' = [32, 24, 12, 13, 31, 33, 14, 21, 34, 22, 23, 11]$ . Finally, according to  $\mathbf{X}'$ ,  $\mathbf{P}$  is recovered as  $\mathbf{P} = [11, 12, 13, 14, 21, 22, 23, 24, 31, 32, 33, 34]$ , which coincides with the original plain image array of Equation (23).

Through the two examples, our attack attains demonstration. Therefore, Pak's encryption scheme cannot resist the chosen-plaintext attacks and the security of the algorithm is not high enough.

#### 4. The Improved Cryptosystem

In Pak's encryption scheme, the diffusion array  $\mathbf{D}'$  and the permutation position array  $\mathbf{X}'$  are used separately in the diffusion and permutation stage. Accordingly, the diffusion array  $\mathbf{D}'$  and the permutation position array  $\mathbf{X}'$  are easily deciphered separately by the attackers. This is a weakness of Pak's encryption scheme. In Wang's improved encryption scheme, a parameter  $E$  determined by the plaintext image is introduced. In order to obtain the value of the  $E$  parameter, it is necessary to calculate the average value of all the pixels of the image, which obviously increases the time overhead of the algorithm. In addition, the linear transformation operation of Wang's algorithm is changed to the binary shift operation to each pixel, which makes encryption speed very slow.

Our improved algorithm retains the advantages of the speed of the original algorithm and overcomes its shortcomings. It includes two rounds of synchronous operations of diffusion and confusion. Two diffusion arrays  $\mathbf{D}'$  and  $\mathbf{D}$  are generated by using the chaotic sequence  $\mathbf{X}$  and the previously encrypted pixel value.  $\mathbf{D}'$  and  $\mathbf{D}$  are used to encrypt the image pixels respectively in the two rounds of synchronous operation.

##### 4.1. Encryption Process

Step 1: Input the secret parameters  $\{x_0, u, k, N_0, C_0\}$  and the color image  $\mathbf{PI}$  with the size of  $M \times N \times 3$ , and  $\mathbf{PI}$  is reshaped to a one-dimensional grayscale image array  $\mathbf{P} = [p(1), p(2), \dots, p(L)]$ , where  $L = M \times N \times 3$ .

Step 2: By using the parameters of  $\{x_0, u, k, N_0\}$ , iterate the new chaotic Sine-Sine system  $(L + N_0)$  times and abandon the front  $N_0$  elements to make the chaotic sequence  $\mathbf{X} = [x(1), x(2), \dots, x(L)]$ .

Step 3: Get the permutation position matrix  $\mathbf{X}' = [x'(1), x'(2), \dots, x'(L)]$  by sorting the chaotic sequence  $\mathbf{X}$  in ascending order. Where,  $x'(i)$  are integers ranging from 1 to  $L$ ,  $i = 1, 2, \dots, L$ .

Step 4: Perform the permutation and diffusion operations on array  $\mathbf{P}$  simultaneously and obtain the temporary cipher image pixel array  $\mathbf{C}' = [c'(1), c'(2), \dots, c'(L)]$  as

$$\begin{cases} d'(1) = \text{mod}(\text{floor}(\frac{x(1)+C_0/256}{2} \times 10^{10}), 256) \\ c'(1) = \text{mod}(p(x'(1)) + d'(1) + C_0, 256) \end{cases} \quad (25)$$

$$\begin{cases} d'(i) = \text{mod}(\text{floor}(\frac{x(i)+c'(i-1)/256}{2} \times 10^{10}), 256), \\ c'(i) = \text{mod}(p(x'(i)) + d'(i) + c'(i-1), 256), i > 1. \end{cases} \quad (26)$$

where,  $\mathbf{D}' = [d'(1), d'(2), \dots, d'(L)]$  is the first diffusion array.

Step 5: Obtain the final cipher image pixel array  $\mathbf{C} = [c(1), c(2), \dots, c(L)]$  from the second diffusion array  $\mathbf{D}$ , permutation position matrix  $\mathbf{X}'$  and the temporary cipher image pixel array  $\mathbf{C}'$  as

$$\begin{cases} d(1) = \text{mod}(\text{floor}(\frac{x(1)+c'(L)/256}{2} \times 10^{10}), 256) \\ c(1) = \text{mod}(c'(1) + d(1) + x'(1) + c'(L), 256) \end{cases} \quad (27)$$

$$\begin{cases} d(i) = \text{mod}(\text{floor}(\frac{x(i)+c'(i-1)/256}{2} \times 10^{10}), 256), \\ c(i) = \text{mod}(c'(i) + d(i) + x'(i) + c(i-1), 256), i > 1. \end{cases} \quad (28)$$

where,  $\mathbf{D} = [d(1), d(2), \dots, d(L)]$  is the second diffusion array.

Step 6: Transform the 1D vector  $\mathbf{C}$  into a 3D matrix with a size of  $M \times N \times 3$ , then the ciphered color image  $\mathbf{CI}$  is obtained.

#### 4.2. Decryption Process

To decrypt the cipher image  $\mathbf{CI}$  with the secret keys  $\{x_0, u, k, N_0, C_0\}$ , the following decryption operations can be executed.

Step 1: Transform the 3D matrix  $\mathbf{CI}$  into a gray scale image pixel sequence  $\mathbf{C}$ .

Step 2: Similar to Step 2 of the encryption process, generate the chaotic sequence  $\mathbf{X} = [x(1), x(2), \dots, x(L)]$ .

Step 3: Similar to Step 3 of the encryption process, get the permutation position matrix  $\mathbf{X}' = [x'(1), x'(2), \dots, x'(L)]$  by sorting  $\mathbf{X}$ .

Step 4: Obtain the temporary cipher image pixel array  $\mathbf{C}' = [c'(1), c'(2), \dots, c'(L)]$  as

$$\begin{cases} d(i) = \text{mod}(\text{floor}(\frac{x(i)+c(i-1)/256}{2} \times 10^{10}), 256), \\ c'(i) = \text{mod}(c(i) - d(i) - x'(i) - c(i-1), 256), i > 1. \end{cases} \quad (29)$$

$$\begin{cases} d(1) = \text{mod}(\text{floor}(\frac{x(1)+c(L)/256}{2} \times 10^{10}), 256) \\ c'(1) = \text{mod}(c(1) - d(1) - x'(1) - c'(L), 256) \end{cases} \quad (30)$$

Step 5: Obtain the recovered plain image pixel array  $\mathbf{P} = [p(1), p(2), \dots, p(L)]$  as

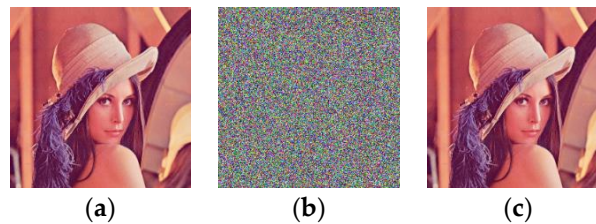
$$\begin{cases} d'(1) = \text{mod}(\text{floor}(\frac{x(1)+C_0/256}{2} \times 10^{10}), 256) \\ p(x'(1)) = \text{mod}(c'(1) - d'(1) - C_0, 256) \end{cases} \quad (31)$$

$$\begin{cases} d'(i) = \text{mod}(\text{floor}(\frac{x(i)+c'(i-1)/256}{2} \times 10^{10}), 256), \\ p(x'(i)) = \text{mod}(c'(i) - d'(i) - c'(i-1), 256), i > 1. \end{cases} \quad (32)$$

Step 6: Transform  $\mathbf{P}$  into a 3D matrix, and the decrypted color image  $\mathbf{PI}$  is obtained.

## 5. Tests and Analysis for the Improved Cryptosystem

To examine the performance of the improved cryptosystem, we carried out a simulation experiment. The secret keys were set as  $(x_0 = 0.4563, u = 5.4321, k = 14, N_0 = 1000, C_0 = 98)$ . The encryption and decryption algorithms were run on the platform Matlab R2016b in a computer with 3.3 GHz CPU, 4 GB memory and a 64 bit Microsoft Windows 7 operating system. The plain image used in the experiments was the color image lena. Figure 6 shows the original plain image and its cipher image encrypted by the improved scheme. The results reveal that the improved scheme has reliable encryption and decryption effect.



**Figure 6.** The encryption and decryption effect of the improved scheme. (a) The plain image; (b) the cipher image; (c) the decrypted image.

### 5.1. Resistance to Chosen-Plaintext Attacks

In our improved scheme, the diffusion matrices  $\mathbf{D}'$  and  $\mathbf{D}$  are related to the temporary and final ciphertext image, which is evident from Equations (25)–(28). It means that images with different contents are encrypted with different diffusion matrices. Furthermore, by using two rounds of diffusion processes, the change of the pixel value at any position in the image will affect all cipher pixel values. Even if the opponent cracked the key streams  $\mathbf{D}'$  and  $\mathbf{D}$  with some specially selected plain images, the key streams  $\mathbf{D}'$  and  $\mathbf{D}$  cannot be used to decrypt the target cipher image because the key streams of the target cipher image are different from the cracked key streams. Moreover, it is difficult to decipher the key streams  $\mathbf{D}'$  and  $\mathbf{D}$  directly by using chosen-plaintext attacks. Therefore, the improved scheme can well resist the chosen-plaintext attacks.

### 5.2. Key Space Analyses

In order to resist a brute-force attack, a cryptographic system must have enough large key space. In our improved cryptosystem, the secret keys include:  $x_0, u, k, N_0, C_0$ , so its key space is  $2^{128}$ , which is the same as those in Reference [60]. Under the current computing power, the key space is large enough to resist a brute-force attack. The size of the key space depends not only on the number of keys but also on the number of possible values for each key. The problem of numerical chaotic systems is that the finite precision of the machines (e.g., computers) leads to performance degradation [63–66], such as the key space is reduced, some weak keys appear, and the randomness of the sequence is reduced. In order to identify and avoid weak keys, we need to calculate the Lyapunov exponents of chaotic systems or plot the phase space trajectories of the system.

### 5.3. Statistical Analysis

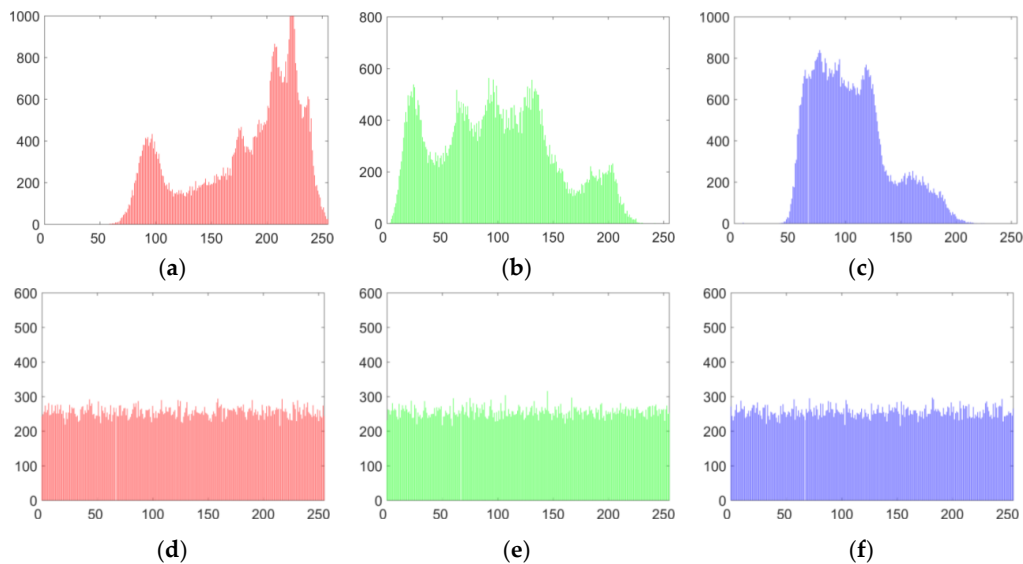
#### 5.3.1. Histogram Analysis

An image histogram displays the distribution of the values of its pixels and provides some statistical information about the image. The histograms of each component of the color lena image and its cipher image are shown in Figure 7. The experimental results in Figure 7 show objectively the statistical distribution of plaintext and ciphertext pixels. The histogram of the cipher image shows that the pixel distribution in the cipher image is very uniform, which means that our improved algorithm has excellent performance in resisting statistical attacks.

The variance of a histogram can quantitatively describe the distribution of pixel values, which is calculated by [54]

$$\text{var}(\mathbf{Z}) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \frac{1}{2} (z_i - z_j)^2. \quad (33)$$

where  $\mathbf{Z}$  is a vector and  $\mathbf{Z} = \{z_1, z_2, \dots, z_{256}\}$ ,  $z_i$  and  $z_j$  are the numbers of pixels with gray values equal to  $i$  and  $j$  respectively. The lower value of variance indicates the higher uniformity of ciphered images.



**Figure 7.** Encryption results for lena. (a) The histograms of R component of Figure 6a; (b) the histograms of G component of Figure 6a; (c) the histograms of B component of Figure 6a; (d) the histograms of R component of Figure 6b; (e) the histograms of G component of Figure 6b; (f) the histograms of B component of Figure 6b.

In the experimental tests, the variances of the histograms of the lena plain image (size of  $256 \times 256 \times 3$ ) and its cipher image were calculated by using Equation (33). The results obtained using two different algorithms are listed in Table 1. From Table 1, one can see that the average variance of the cipher image lena obtained with the proposed improved algorithm is 241.4141, which is much less than that of Wang's algorithm [61]. Thus, our improved algorithm has better performance in resisting statistical attacks.

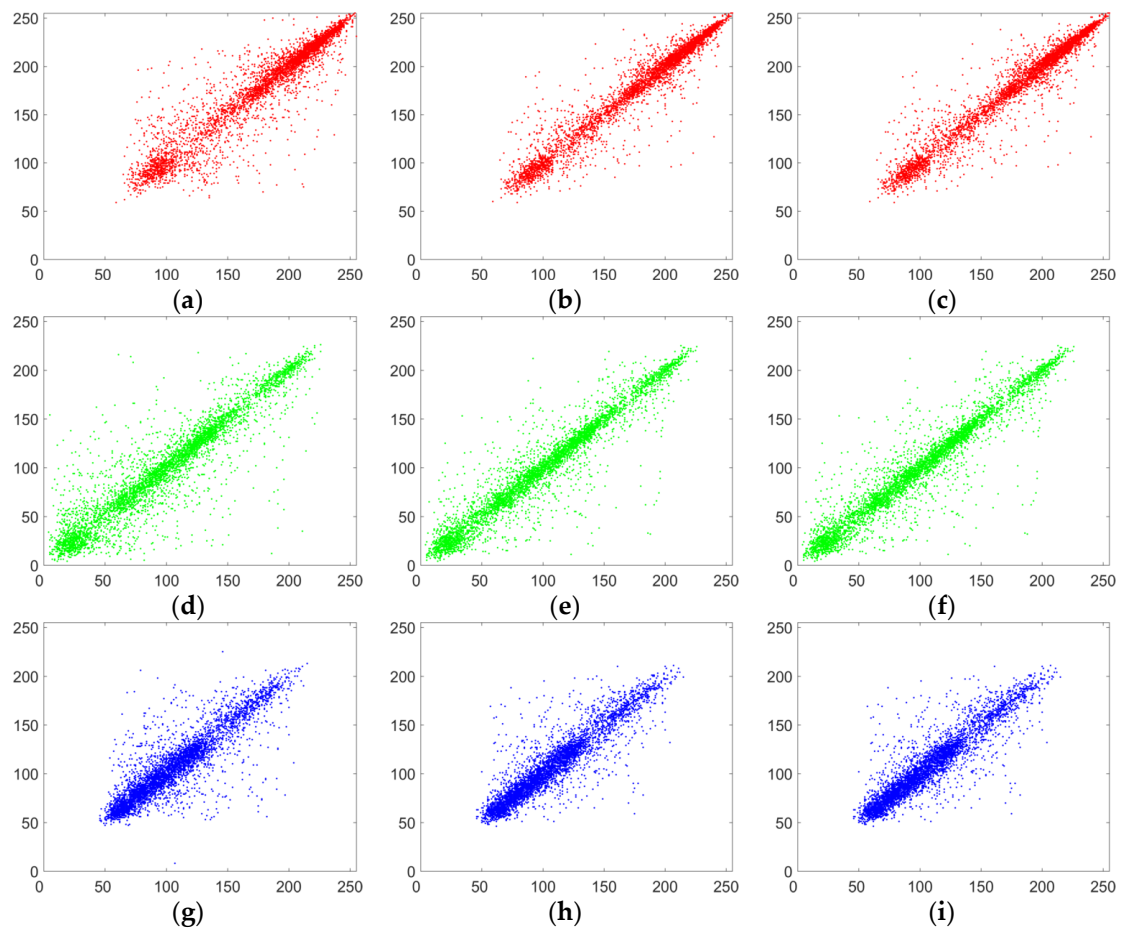
**Table 1.** Variances of the histograms of the Lena image.

Channel	Plain Image	Cipher Image [61]	Cipher Image
R	63,888.1328	527.3242	244.6797
G	28,546.0078	504.7522	239.7656
B	86,487.8906	501.6874	239.7969
Average	57,516.9492	511.2546	241.4141

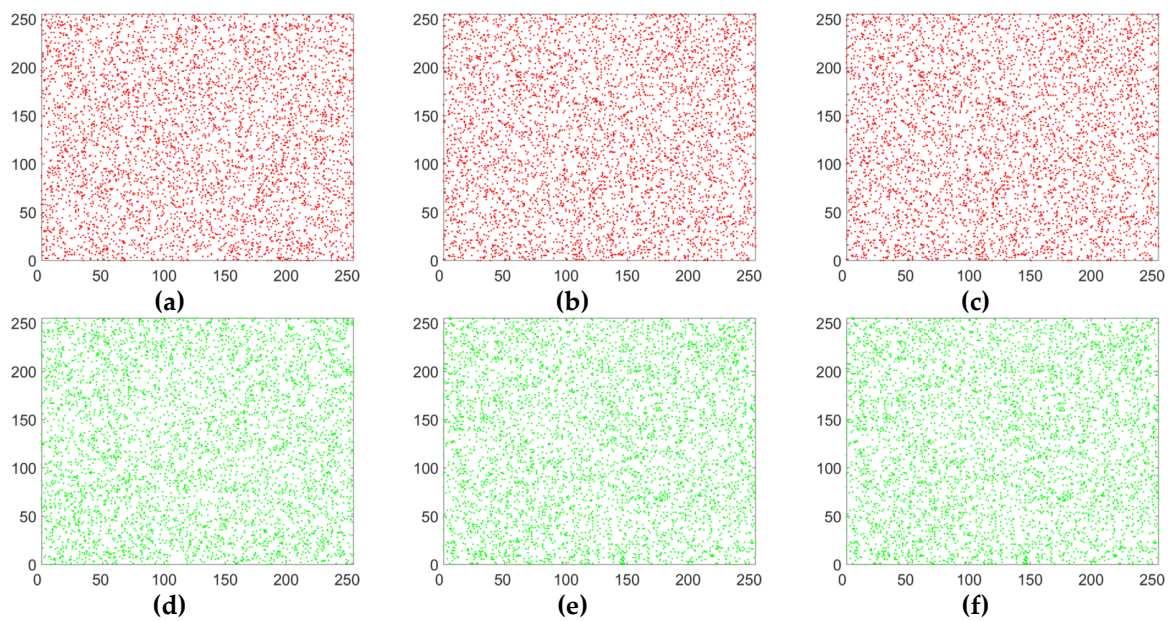
### 5.3.2. Correlation of Two Adjacent Pixels

Adjacent pixels in images usually have a strong correlation. A good encryption algorithm should break the correlation of adjacent pixels in an image. In order to directly describe the correlation of adjacent pixels in an image, based on 5000 randomly selected pairs of pixels (in horizontal, vertical and diagonal directions), the correlation distribution graphs of the lena plain image and its corresponding cipher image are drawn in Figures 8 and 9. It can be seen that the adjacent pixels in three directions in the plain image have a strong correlation, while those in the cipher image have almost no correlation and it is a random pattern. The results mean that our improved scheme has greatly eliminated the correlation of adjacent pixels.



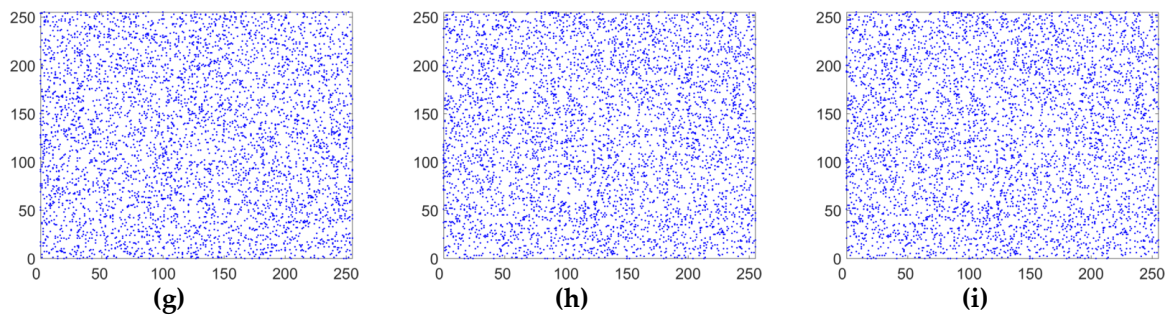


**Figure 8.** Correlation analysis of the plain image. (a) Horizontal correlation in R channel; (b) vertical correlation in R channel; (c) diagonal correlation in R channel; (d) horizontal correlation in G channel; (e) vertical correlation in G channel; (f) diagonal correlation in G channel; (g) horizontal correlation in B channel; (h) vertical correlation in B channel; (i) diagonal correlation in B channel.



**Figure 9.** Cont.





**Figure 9.** Correlation analysis of the corresponding cipher image. (a) Horizontal correlation in R channel; (b) vertical correlation in R channel; (c) diagonal correlation in R channel; (d) horizontal correlation in G channel; (e) vertical correlation in G channel; (f) diagonal correlation in G channel; (g) horizontal correlation in B channel; (h) vertical correlation in B channel; (i) diagonal correlation in B channel.

In order to quantitatively depict the correlation of adjacent pixels of an image, we introduce correlation coefficient index  $r_{XY}$ , which is calculated as follows:

$$r_{XY} = \text{cov}(X, Y) / \sqrt{D(X)}\sqrt{D(Y)} \tag{34}$$

$$E(X) = \frac{1}{N} \sum_{i=1}^N x_i \tag{35}$$

$$D(X) = \frac{1}{N} \sum_{i=1}^N (x_i - E(X))^2 \tag{36}$$

$$\text{cov}(X, Y) = \frac{1}{N} \sum_{i=1}^N (x_i - E(X))(y_i - E(Y)). \tag{37}$$

where  $X$  and  $Y$  are gray-scale values of two adjacent pixels in the images. For the color lena image, the correlation coefficients of adjacent pixels in R component of plaintext image and R component of ciphertext image were calculated respectively. The results are listed in Table 2. From Table 2, we can see that the correlation coefficients of adjacent pixels in R component of plaintext image are close to 1 while those of the cipher image are close to 0. The experimental results also show that our improved algorithm has smaller absolute values of correlation coefficient than Wang’s algorithm in the vertical and diagonal directions and Pak’s algorithm in all three directions.

**Table 2.** Correlation coefficients of the plain image and cipher images of lena in the R channel.

Directions	Plain Image	Cipher Image		
		R	Reference [61]	Ours
H	0.9567	−0.0026	0.00037	0.00063
V	0.9239	−0.0038	−0.00540	−0.00052
D	0.8888	0.0017	0.00166	−0.00012

### 5.3.3. Sensitivity Analysis

In order to resist differential attacks, the algorithm must be sensitive to the secret keys and plain images. To measure the sensitivity of an algorithm to tiny changes in key or plain image, we cite two

metrics. One is the number of pixel changing rate (NPCR), another is the unified averaged changed intensity (UACI). The definitions of NPCR and UACI are

$$\text{NPCR} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \delta(i, j) \times 100\%, \quad (38)$$

$$\text{UACI} = \frac{1}{m \times n} \left( \sum_{i=1}^m \sum_{j=1}^n \frac{|c_1(i, j) - c_2(i, j)|}{255} \right) \times 100\%. \quad (39)$$

where  $m, n$  represent the pixel rows and columns of an image, respectively. Here,  $C_1 = [c_1(i, j)]$  and  $C_2 = [c_2(i, j)]$  express two encrypted images corresponding to two security keys or two plain images, and  $\delta(i, j)$  is computed by

$$\delta(i, j) = \begin{cases} 1, & \text{if } c_1(i, j) \neq c_2(i, j), \\ 0, & \text{if } c_1(i, j) = c_2(i, j). \end{cases} \quad (40)$$

The desired value of NPCR is 1 and the desired value of UACI is 0.3346 [54].

To measure the sensitivity of our improved algorithm for the plain image, the color lena image (size  $256 \times 256 \times 3$ ) is chosen as the plain image one, and the plain image two is obtained by changing only one pixel of the plain image one. Then, two encrypted images are obtained by executing the improved encryption algorithm with the same secret keys, respectively. NPCR and UACI values are computed with two cipher images, and the results are listed in Table 3. The results indicate that our improved encryption algorithm is very sensitive to the plain image.

**Table 3.** Values of NPCR and UACI of Lena cipher images.

Channel	NPCR [61]	NPCR	UACI [61]	UACI
R	0.996413	1	0.334801	0.3341
G	0.996328	1	0.334791	0.3363
B	0.996250	0.9974	0.334558	0.3346

To measure the sensitivity of the improved algorithm to the secret keys, two different keys with a tiny difference are used to encrypt the same plain image lena and the two cipher images,  $C_1$  and  $C_2$ , are obtained. The tiny change ( $10^{-14}$ ) is introduced to one of the secret keys ( $x_0, u$ ) while keeping all the others unchanged. Similarly,  $k$  is changed to  $k + 1$ ,  $N_0$  is changed to  $N_0 + 1$ ,  $C_0$  is changed to  $C_0 + 1$ , while keeping all the others unchanged. The NPCR and UACI of the cipher images  $C_1$  and  $C_2$  are given in Tables 4 and 5. The experimental results indicate that our improved algorithm is very sensitive to any slight change in each secret key.

**Table 4.** NPCR of the improved algorithm with a slight change in the secret keys.

Channel	$x_0 + 10^{-14}$	$u + 10^{-14}$	$k + 1$	$N_0 + 1$	$C_0 + 1$
R	0.9961	0.9960	0.9958	0.9959	0.9961
G	0.9959	0.9964	0.9962	0.9962	0.9961
B	0.9963	0.9961	0.9960	0.9964	0.9961

**Table 5.** UACI of the improved algorithm with a slight change in the secret keys.

Channels	$x_0 + 10^{-14}$	$u + 10^{-14}$	$k + 1$	$N_0 + 1$	$C_0 + 1$
R	0.3334	0.3356	0.3353	0.3370	0.3344
G	0.3350	0.3355	0.9962	0.3348	0.3348
B	0.3343	0.3355	0.3352	0.3340	0.3337

### 5.3.4. Information Entropy Analysis

Image information entropy is an important way to measure the randomness of the pixel distribution. Let  $I$  be an image and its information entropy can be calculated as:

$$H(I) = - \sum_{i=0}^{2^n-1} P(I_i) \log_2[P(I_i)], \quad (41)$$

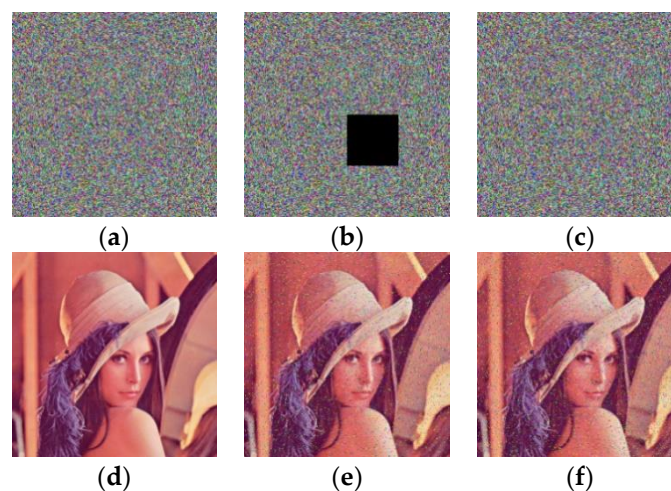
where  $P(I_i)$  denotes the occurrence probability of gray level  $i$ ,  $I_i = i$ , and  $i = 0, 1, 2, \dots, 2^n$ . Here,  $2^n$  is the number of grayscale levels of an image. If  $P(I_i) = 1/2^n$ , then the image is completely random. For an image with 256 gray-scales,  $n = 8$  and the image has  $2^8$  grayscale levels, so the ideal value of information entropy is 8. For an encrypted image, the closer the entropy is to 8, the closer the image is to a randomly distributed image. We experimentally tested the information entropy of the color lena ciphertext images obtained by three kinds of encryption algorithms. The results of the information entropy corresponding to the R, G and B channels are listed in Table 6. From Table 6, one can see that all the entropy values are significantly closer to 8, so the randomness is satisfactory. Among these three algorithms, our improved algorithm has the largest average entropy value. Hence, our improved encryption scheme is more capable of resisting information entropy-based attacks.

**Table 6.** Entropies of the encrypted lena image by three encryption schemes.

Channels	Reference [60]	Reference [61]	Ours
R	7.9971	7.9970	7.9973
G	7.9972	7.9965	7.9973
B	7.9974	7.9973	7.9974
Average	7.9972	7.9969	7.9973

### 5.3.5. Cropping and Noise Attack

To test the performance of our improved scheme in resisting data loss and noise attacks. The encrypted lena image (Figure 10a) was attacked by a data cut with a size of  $64 \times 64$  (Figure 10b) and a 3% “salt & pepper” noise attack (Figure 10c), respectively. Then, these cipher images were decrypted respectively and the results of the decryption are given in Figure 10d–f. The results indicate that our improved scheme can resist cutting and noise pollution attacks.



**Figure 10.** Data loss and noise attack. (a) The original cipher image; (b) the cipher images with data loss; (c) the cipher image added with 3% “salt & pepper” noise; (d) the decrypted image of (a); (e) the decrypted image of (b); (f) the decrypted image of (c).

#### 5.4. Analysis of Speed

A practical encryption algorithm should be efficient in terms of encryption speed. To test the encryption speed of the improved scheme, three RGB color images with different size have been used for the encryption. The simulation experiments were run on a desktop PC with Intel(R) Core i5-4590 3.30 GHz CPU, 4 GB RAM and 500 GB hard disk. The operating system was 64 bits Microsoft Windows 7 and the computational platform was Matlab R2016b. The average encryption/decryption time taken by Pak's algorithm, Wang's algorithm and our improved algorithm for processing the images with different size are shown in Table 7. The results show that our algorithm has the fastest speed. This is because our encryption algorithm has abandoned binary XOR operations.

**Table 7.** The time cost tests.

Image size	Reference [60]	Reference [61]	Ours
256 × 256	0.5693 s	8.2328 s	0.3873 s
512 × 512	2.2340 s	32.7673 s	1.5145 s
1024 × 1024	8.9055 s	131.6625 s	6.0163 s

## 6. Conclusions

In this paper, an improved cryptanalysis on a color image cryptosystem is presented. It has been shown that the equivalent secret key and all the unknown parameters of the cryptosystem can be recovered by our chosen-plaintext attack algorithm. Furthermore, based on the analysis of defects in the original cryptosystem, an improved color image encryption scheme is proposed. The contributions of this paper include two aspects: First, a more complete and efficient method to comprehensively crack Pak's encryption scheme is proposed, which further enriches the research of cryptanalysis. The validity and correctness of the cryptanalysis algorithm were verified by theoretical analysis and experimental results. Second, a new color image encryption algorithm with a higher security and a higher encryption efficiency is proposed. In the new encryption scheme, the generation of diffusion arrays depends on the content of the image itself and the permutation position array. In the process of diffusion, two effects of ciphertext feedback and pixel scrambling are also implemented simultaneously. Using these methods, the security of the cryptosystem is enhanced. Experimental results and security analysis demonstrate that the improved cryptosystem can achieve a satisfactory security level after two rounds of diffusion encryption.

Looking to the future in image encryption field, some new research directions are worth considering, such as efficient image encryption technology in the resource-constrained mobile social network [67] or sensor network communication environment [68]. Another interesting form of encryption is searchable encryption [69], which is a very promising direction in the field of cloud computing.

**Author Contributions:** Conceptualization, C.Z. and G.W.; methodology, C.Z.; software, C.Z.; validation, C.Z., G.W. and K.S.; formal analysis, C.Z.; investigation, C.Z.; resources, C.Z.; data curation, K.S.; writing—original draft preparation, C.Z.; writing—review and editing, G.W.; visualization, K.S.; supervision, G.W.; project administration, K.S.; funding acquisition, G.W.

**Funding:** This research was funded by [the Open Project of Guangxi Colleges and Universities Key Laboratory of Complex System Optimization and Big Data Processing] grant number [No. 2016CSOBDP0103]; [the National Natural Science Foundation of China] grant number [Nos. 61472451 and 61632009].

**Acknowledgments:** The authors are thankful to the reviewers for their comments and suggestions to improve the quality of the manuscript.

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

## References

1. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [[CrossRef](#)]
2. Zanette, D.H. Generalized kolmogorov entropy in the dynamics of the multifractal generation. *Phys. A Stat. Mech. Appl.* **1996**, *223*, 87–98. [[CrossRef](#)]
3. Crutchfield, J.P.; Packard, N.H. Symbolic dynamics of noisy chaos. *Phys. D* **1983**, *7*, 201–223. [[CrossRef](#)]
4. Crutchfield, J.P.; Feldman, D.P. Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos* **2003**, *13*, 25–54. [[CrossRef](#)] [[PubMed](#)]
5. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284. [[CrossRef](#)]
6. Zhang, Y.; Xiao, D. An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 74–82. [[CrossRef](#)]
7. Zhang, Y.; Xiao, D. Double optical image encryption using discrete chirikov standard map and chaos-based fractional random transform. *Opt. Lasers Eng.* **2013**, *51*, 472–480. [[CrossRef](#)]
8. Gan, Z.H.; Chai, X.L.; Han, D.J.; Chen, Y.R. A chaotic image encryption algorithm based on 3-D bit-plane permutation. *Neural Comput. Appl.* **2018**, 1–20. [[CrossRef](#)]
9. Hu, G.; Xiao, D.; Zhang, Y.; Xiang, T. An efficient chaotic image cipher with dynamic lookup table driven bit-level permutation strategy. *Nonlinear Dyn.* **2016**, *87*, 1359–1375. [[CrossRef](#)]
10. Ye, G.; Zhao, H.; Chai, H. Chaotic image encryption algorithm using wave-line permutation and block diffusion. *Nonlinear Dyn.* **2016**, *83*, 2067–2077. [[CrossRef](#)]
11. Abd-El-Hafiz, S.K.; AbdElHaleem, S.H.; Radwan, A.G. Novel permutation measures for image encryption algorithms. *Opt. Lasers Eng.* **2016**, *85*, 72–83. [[CrossRef](#)]
12. 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. [[CrossRef](#)]
13. Zhang, Y.; Xiao, D.; Shu, Y.; Li, J. A novel image encryption scheme based on a linear hyperbolic chaotic system of partial differential equations. *Signal Process. Image Commun.* **2013**, *28*, 292–300. [[CrossRef](#)]
14. Wang, X.; Liu, C.; Zhang, H. An effective and fast image encryption algorithm based on chaos and interweaving of ranks. *Nonlinear Dyn.* **2016**, *84*, 1595–1607. [[CrossRef](#)]
15. Xu, L.; Gou, X.; Li, Z.; Li, J. A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion. *Opt. Lasers Eng.* **2017**, *91*, 41–52. [[CrossRef](#)]
16. Hua, Z.; Yi, S.; Zhou, Y. Medical image encryption using high-speed scrambling and pixel adaptive diffusion. *Signal Process.* **2018**, *144*, 134–144. [[CrossRef](#)]
17. Huang, H.; He, X.; Xiang, Y.; Wen, W.; Zhang, Y. A compression-diffusion-permutation strategy for securing image. *Signal Process.* **2018**, *150*, 183–190. [[CrossRef](#)]
18. Cao, C.; Sun, K.; Liu, W. A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. *Signal Process.* **2018**, *143*, 122–133. [[CrossRef](#)]
19. Chai, X. An image encryption algorithm based on bit level brownian motion and new chaotic systems. *Multimed. Tools Appl.* **2017**, *76*, 1159–1175. [[CrossRef](#)]
20. Hua, Z.; Jin, F.; Xu, B.; Huang, H. 2D Logistic-Sine-coupling map for image encryption. *Signal Process.* **2018**, *149*, 148–161. [[CrossRef](#)]
21. Hua, Z.; Zhou, Y. Image encryption using 2D Logistic-adjusted-Sine map. *Inf. Sci.* **2016**, *339*, 237–253. [[CrossRef](#)]
22. Kaur, M.; Kumar, V. Efficient image encryption method based on improved lorenz chaotic system. *Electron. Lett.* **2018**, *54*, 562–564. [[CrossRef](#)]
23. Liu, J.; Yang, D.; Zhou, H.; Chen, S. A digital image encryption algorithm based on bit-planes and an improved logistic map. *Multimed. Tools Appl.* **2018**, *77*, 10217–10233. [[CrossRef](#)]
24. Zhu, C. A novel image encryption scheme based on improved hyperchaotic sequences. *Opt. Commun.* **2012**, *285*, 29–37. [[CrossRef](#)]
25. Zhang, Y.; Tang, Y. A plaintext-related image encryption algorithm based on chaos. *Multimed. Tools Appl.* **2018**, *77*, 6647–6669. [[CrossRef](#)]
26. Ye, G.; Huang, X. A secure image encryption algorithm based on chaotic maps and SHA-3. *Secur. Commun. Netw.* **2016**, *9*, 2015–2023. [[CrossRef](#)]



27. Wu, X.; Kan, H.; Kurths, J. A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps. *Appl. Soft Comput.* **2015**, *37*, 24–39. [[CrossRef](#)]
28. Wang, X.; Zhu, X.; Wu, X.; Zhang, Y. Image encryption algorithm based on multiple mixed hash functions and cyclic shift. *Opt. Lasers Eng.* **2017**, *107*, 370–379. [[CrossRef](#)]
29. Chai, X.; Gan, Z.; Zhang, M. A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion. *Multimed. Tools Appl.* **2016**, *76*, 15561–15585. [[CrossRef](#)]
30. Chai, X.; Chen, Y.; Broyde, L. A novel chaos-based image encryption algorithm using DNA sequence operations. *Opt. Lasers Eng.* **2017**, *88*, 197–213. [[CrossRef](#)]
31. Guesmi, R.; Farah, M.A.B.; Kachouri, A.; Samet, M. A novel chaos-based image encryption using DNA sequence operation and secure hash algorithm SHA-2. *Nonlinear Dyn.* **2016**, *83*, 1123–1136. [[CrossRef](#)]
32. Hu, T.; Liu, Y.; Gong, L.-H.; Guo, S.-F.; Yuan, H.-M. Chaotic image cryptosystem using DNA deletion and DNA insertion. *Signal Process.* **2017**, *134*, 234–243. [[CrossRef](#)]
33. Wang, X.; Liu, C. A novel and effective image encryption algorithm based on chaos and DNA encoding. *Multimed. Tools Appl.* **2016**, *76*, 6229–6245. [[CrossRef](#)]
34. Wang, X.-Y.; Li, P.; Zhang, Y.-Q.; Liu, L.-Y.; Zhang, H.; Wang, X. A novel color image encryption scheme using DNA permutation based on the Lorenz system. *Multimed. Tools Appl.* **2017**, *77*, 6243–6265. [[CrossRef](#)]
35. Wang, X.-Y.; Zhang, Y.-Q.; Bao, X.-M. A novel chaotic image encryption scheme using DNA sequence operations. *Opt. Lasers Eng.* **2015**, *73*, 53–61. [[CrossRef](#)]
36. Zhang, L.-M.; Sun, K.-H.; Liu, W.-H.; He, S.-B. A novel color image encryption scheme using fractional-order hyperchaotic system and DNA sequence operations. *Chin. Phys. B* **2017**, *26*, 100504. [[CrossRef](#)]
37. Zhang, D.; Liao, X.; Yang, B.; Zhang, Y. A fast and efficient approach to color-image encryption based on compressive sensing and fractional Fourier transform. *Multimed. Tools Appl.* **2018**, *77*, 2191–2208. [[CrossRef](#)]
38. Wang, X.; Wang, Q.; Zhang, Y. A fast image algorithm based on rows and columns switch. *Nonlinear Dyn.* **2015**, *79*, 1141–1149. [[CrossRef](#)]
39. Tong, X.-J.; Zhang, M.; Wang, Z.; Liu, Y.; Xu, H.; Ma, J. A fast encryption algorithm of color image based on four-dimensional chaotic system. *J. Vis. Commun. Image Represent.* **2015**, *33*, 219–234. [[CrossRef](#)]
40. Liu, H.; Kadir, A.; Sun, X. Chaos-based fast colour image encryption scheme with true random number keys from environmental noise. *IET Image Process.* **2017**, *11*, 324–332. [[CrossRef](#)]
41. Liu, W.; Sun, K.; Zhu, C. A fast image encryption algorithm based on chaotic map. *Opt. Lasers Eng.* **2016**, *84*, 26–36. [[CrossRef](#)]
42. Bibi, N.; Farwa, S.; Muhammad, N.; Jahngir, A.; Usman, M. A novel encryption scheme for high-contrast image data in the Fresnel domain. *PLoS ONE* **2018**, *13*, e0194343.
43. Farwa, S.; Muhammad, N.; Shah, T.; Ahmad, S. A novel image encryption based on algebraic s-box and Arnold transform. *3D Res.* **2017**, *8*, 26. [[CrossRef](#)]
44. Farwa, S.; Shah, T.; Muhammad, N.; Bibi, N.; Jahangir, A.; Arshad, S. An image encryption technique based on chaotic s-box and Arnold transform. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 360–364. [[CrossRef](#)]
45. Martin, K.; Lukac, R.; Plataniotis, K.N. Efficient encryption of wavelet-based coded color images. *Pattern Recognit.* **2005**, *38*, 1111–1115. [[CrossRef](#)]
46. Shahed, M.A. Wavelet based fast technique for images encryption. *Basrah J. Sci.* **2007**, *25*, 126–141.
47. Gao, H.J.; Zhang, Y.S.; Liang, S.Y.; Li, D.Q. A new chaotic algorithm for image encryption. *Chaos Solitons Fractals* **2006**, *29*, 393–399. [[CrossRef](#)]
48. Guariglia, E. Entropy and fractal antennas. *Entropy* **2016**, *18*, 84. [[CrossRef](#)]
49. Guariglia, E. Harmonic Sierpinski gasket and applications. *Entropy* **2018**, *20*, 714. [[CrossRef](#)]
50. Li, C.; Lin, D.; Lu, J. Cryptanalyzing an image-scrambling encryption algorithm of pixel bits. *IEEE Multimed.* **2017**, *24*, 64–71. [[CrossRef](#)]
51. Li, C.; Liu, Y.; Xie, T.; Chen, M.Z.Q. Breaking a novel image encryption scheme based on improved hyperchaotic sequences. *Nonlinear Dyn.* **2013**, *73*, 2083–2089. [[CrossRef](#)]
52. Wang, X.; Luan, D.; Bao, X. Cryptanalysis of an image encryption algorithm using Chebyshev generator. *Digit. Signal Process.* **2014**, *25*, 244–247. [[CrossRef](#)]
53. Wu, J.; Liao, X.; Yang, B. Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation. *Signal Process.* **2018**, *142*, 292–300. [[CrossRef](#)]
54. Zhu, C.; Sun, K. Cryptanalyzing and improving a novel color image encryption algorithm using rt-enhanced chaotic tent maps. *IEEE Access* **2018**, *6*, 18759–18770. [[CrossRef](#)]

55. Wu, X.; Zhu, B.; Hu, Y.; Ran, Y. A novel colour image encryption scheme using rectangular transform-enhanced chaotic tent maps. *IEEE Access* **2017**, *5*, 6429–6436. [[CrossRef](#)]
56. Zhang, W.; Yu, H.; Zhao, Y.L.; Zhu, Z.L. Image encryption based on three-dimensional bit matrix permutation. *Signal Process.* **2016**, *118*, 36–50. [[CrossRef](#)]
57. Sun, K.H.; He, S.B.; Yin, L.Z.; Duo, L.K. Application of fuzzyen algorithm to the analysis of complexity of chaotic sequence. *Acta Phys. Sin.* **2012**, 130507.
58. Sun, K.H.; He, S.B.; He, Y.; Yin, L.Z. Complexity analysis of chaotic pseudo-random sequences based on spectral entropy algorithm. *Acta Phys. Sin.* **2013**, *62*, 010501.
59. He, S.B.; Sun, K.H.; Zhu, C.X. Complexity analyses of multi-wing chaotic systems. *Chin. Phys. B* **2013**, 220–225. [[CrossRef](#)]
60. Pak, C.; Huang, L. A new color image encryption using combination of the 1d chaotic map. *Signal Process.* **2017**, *138*, 129–137. [[CrossRef](#)]
61. Wang, H.; Xiao, D.; Chen, X.; Huang, H. Cryptanalysis and enhancements of image encryption using combination of the 1d chaotic map. *Signal Process.* **2018**, *144*, 444–452. [[CrossRef](#)]
62. Chen, J.; Han, F.; Qian, W.; Yao, Y.-D.; Zhu, Z.L. Cryptanalysis and improvement in an image encryption scheme using combination of the 1d chaotic map. *Nonlinear Dyn.* **2018**, *93*, 2399–2413. [[CrossRef](#)]
63. Li, S.; Chen, G.; Mou, X. On the dynamical degradation of digital piecewise linear chaotic maps. *Int. J. Bifurc. Chaos* **2015**, *15*, 3119–3151. [[CrossRef](#)]
64. Li, S.; Chen, G.; Wong, K.-W.; Mou, X.; Cai, Y. Baptista-type chaotic cryptosystems: Problems and countermeasures. *Phys. Lett. A* **2004**, *332*, 368–375. [[CrossRef](#)]
65. Curiac, D.I.; Volosencu, C. Chaotic trajectory design for monitoring an arbitrary number of specified locations using points of interest. *Math. Probl. Eng.* **2012**, *2012*, 940276. [[CrossRef](#)]
66. Curiac, D.I.; Iercan, D.; Dragan, F.; Baniias, O. Chaos-based cryptography: End of the road? In *Proceedings of the International Conference on Emerging Security Information, System and Technologies*, Valencia, Spain, 14–20 October 2007; pp. 71–76.
67. Zhang, S.; Wang, G.; Liu, Q.; Abawajy, J.H. A trajectory privacy-preserving scheme based on query exchange in mobile social networks. *Soft Comput.* **2018**, *22*, 6121–6133. [[CrossRef](#)]
68. Bhuiyan, M.Z.A.; Wang, G.; Wu, J.; Cao, J.; Liu, X.; Wang, T. Dependable structural health monitoring using wireless sensor networks. *IEEE Trans. Dependable Secur.* **2017**, *14*, 363–376. [[CrossRef](#)]
69. Zhang, Q.; Liu, Q.; Wang, G. PRMS: A personalized mobile search over encrypted outsourced data. *IEEE Access* **2018**, *6*, 31541–31552. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).