entropy

MDPI

Article

On the Security of a Latin-Bit Cube-Based Image Chaotic Encryption Algorithm

Zeqing Zhang * and Simin Yu

School of Automation, Guangdong University of Technology, Guangzhou 510006, China; siminyu@163.com

* Correspondence: 2111704029@mail2.gdut.edu.cn

Received: 15 August 2019; Accepted: 9 September 2019; Published: 12 September 2019

Abstract: In this paper, the security analysis of an image chaotic encryption algorithm based on Latin cubes and bit cubes is given. The proposed algorithm adopts a first-scrambling-diffusion-second-scrambling three-stage encryption scheme. First, a finite field is constructed using chaotic sequences. Then, the Latin cubes are generated from finite field operation and used for image chaotic encryption. In addition, according to the statistical characteristics of the diffusion image in the diffusion stage, the algorithm also uses different Latin cube combinations to scramble the diffusion image for the second time. However, the generation of Latin cubes in this algorithm is independent of plain image, while, in the diffusion stage, when any one bit in the plain image changes, the corresponding number of bits in the cipher image follows the change with obvious regularity. Thus, the equivalent secret keys can be obtained by chosen plaintext attack. Theoretical analysis and experimental results indicate that only a maximum of $2.5 \times \sqrt[3]{w \times h} + 6$ plain images are needed to crack the cipher image with $w \times h$ resolution. The size of equivalent keys deciphered by the method proposed in this paper are much smaller than other general methods of cryptanalysis for similar encryption schemes.

Keywords: image chaotic encryption; cryptography; Latin cube; bit cube; chosen plaintext attack

1. Introduction

Image chaotic encryption algorithms have attracted some special attention in the field of information security [1–7]. In recent years, many image chaotic encryption schemes combined chaos theories with other technologies, such as one-time keys [8], bit-level permutation [9], DNA operations [10–13], parallel computing system [14], matrix semi-tensor product theory [15], cellular automata [16,17], neural network [18,19], Latin square or Latin cube [20–22], and so on, have been proposed. However, the security issues of image chaotic encryption algorithms have also attracted much attention. As a basic requirement of security, the ciphertext image of the image chaotic encryption algorithm must have good uniformity. In addition, the algorithm must have a large enough key space to resist brute force attacks. For instance, in order to show the security of the image chaotic encryption algorithm in the statistical sense, the key space analysis, statistical analysis, and differential analysis of the chaos encryption algorithm proposed in [23] and its corresponding extended algorithm are given in Sections 4 and 5 of [23], respectively. However, the high uniformity of ciphertext does not mean that the encryption algorithm has high security performance. For example, in [24], the security analysis of an image chaotic encryption algorithm proposed in [16] is given, and it is found that the generation of key stream is related to the sum of pixel values of plain images. Under the premise of satisfying the sum of pixel values of a plain image unchanged, only two pixel values of cipher image are changed corresponding to the variation of two pixel values of a plain image, which is vulnerable to differential attack. Therefore, the equivalent secret keys can be obtained by selecting 512 plain images. In [25], the cryptanalysis of a DNA encoding-based image scrambling and diffusion encryption

algorithm proposed in [10] is reported to find that the scrambling algorithm is also independent of plain image, so that it can be deciphered by chosen plaintext attack. In addition, by choosing some specific plain images, the original image chaotic encryption algorithm can be simplified into scrambling-only encryption algorithm, which has been proven to be insecure [26,27]. In [28], the security analysis of an image encryption algorithm based on a compound chaotic system proposed in [29] is given, and it is pointed out that there are a large number of equivalent secret keys in the image chaotic encryption algorithm. In [30], an 8D self-synchronous and feedback-based chaotic stream cipher using the lower 8 bits of one state variable for encryption is proposed. However, in [31], most of the secret keys are successfully acquired by means of a divide and conquer attack, known plaintext attack, and a chosen ciphertext attack, respectively. In [32], the security analysis of a Latin square based image chaotic encryption algorithm proposed in [22] is given to find the security vulnerabilities both in the diffusion stage and in the scrambling stage through chosen text attack. In [33], the chosen plaintext attack is adopted for the safety performance assessment of a 1D combinatorial chaotic encryption algorithm proposed in [34]. In addition, in [35], the chosen plaintext attack is also utilized for analyzing the security of a bit cube-based image chaotic encryption algorithm proposed in [36]. In addition, some chaotic cipher designers have also discovered the importance of cryptanalysis. For example, in Section 3 of [37], the resistance to the four classic attack methods is analyzed in detail. The analysis shows that the proposed encryption algorithm has resistance to the chosen plaintext attack because it is sensitive to the initial parameters.

In 2019, an image chaotic encryption algorithm based on orthogonal Latin cubes and bit cubes is given in [20]. First, a chaotic sequence is generated by logistic mapping, and it is further arranged in ascending order to obtain its corresponding chaotic index sequence. Next, a finite field is constructed by the chaotic index sequence, and three orthogonal Latin cubes are also generated. Then, the generated three orthogonal Latin cubes are used for the first-scrambling-diffusion- second-scrambling three-stage encryption. Although the designer claims that the algorithm has passed various statistical tests, the analysis results in this paper demonstrate that the algorithm has at least two security vulnerabilities as follows:

(1) The generation of Latin cubes in this algorithm is independent of plain image.
(2) When any one bit in the plain image changes, the corresponding number of bits in the cipher image follows the change with obvious regularity.

Based on the above-mentioned security vulnerabilities, this paper adopts both chosen plaintext attack and differential attack for analyzing the safety performance for the image chaotic encryption algorithm proposed in [20]. First, a full zero plain image and multiple non-full zero plain images are selected, and the differential operation is performed between the cipher image corresponding to this full zero plain image and the cipher image corresponding to those non-full zero plain images. On the premise that the sum of bit 1 in each differential operation is even, the chaotic index sequence lx can be deciphered. Next, based on the obtained lx, and on the condition that there exists an intersection in the solutions of unary quadratic equation on finite field $GF(q)$, the secret keys α, β, γ can be further deciphered.

The rest of the paper is organized as follows: Section 2 briefly introduces the image chaotic encryption algorithm. Section 3 presents the security analysis. Section 4 gives the steps for deciphering image chaotic encryption algorithm. Section 5 demonstrates the numerical simulation experiments. Section 6 gives some improvement suggestions for the image chaotic encryption algorithm. Finally, Section 7 concludes the paper.

2. Description of an Image Chaotic Encryption Algorithm

2.1. A Brief View of an Image Chaotic Encryption Algorithm

In [20], the image chaotic encryption algorithm consists of secret keys selection, Latin cube generation, scrambling encryption, and diffusion encryption, as shown in Figure 1, where key_0, μ_0,

α, β, γ are the secret keys, x_n $(n = 0, 1, 2, \cdots)$ is a chaotic sequence generated by Logistic mapping, lx is a chaotic index sequence, L_1, L_2, L_3 are three Latin cubes, P is a 2D plain gray image, M is a bit cube representation of P, S_1 is a first-scrambling image of M, D is a diffusion image of S_1, S_2 is a second-scrambling image of D, E is a 2D cipher gray image of S_2, and B is generated by L_1. When the size of the image is $w \times h$, the length of x_n and lx is $q = \sqrt[3]{8 \times w \times h}$, the side length of Latin cubes and bit cubes is $q = \sqrt[3]{8 \times w \times h}$, and the secret keys α, β, $\gamma \in \{0, 1, 2, \cdots, q-1\}$. Note that an appropriate image size $w \times h$ should be selected to ensure that $q = \sqrt[3]{8 \times w \times h} = 2 \times \sqrt[3]{w \times h}$ is an even number. In Figure 1, L_1, L_2, $L_3 \in \{0, 1, 2, \cdots, q-1\}$ are Latin cubes, M, S_1, D, S_2, $B \in \{0, 1\}$ are bit cubes, P is a 2D plain gray image, E is a 2D cipher gray image, p_k, p_t, s_1, b, d, $e_k \in \{0, 1\}$ are 1D bit sequences corresponding to P, S_1, B, D, E, and $t = T(k)$ is a position scrambling rule corresponding to the first-scrambling stage.

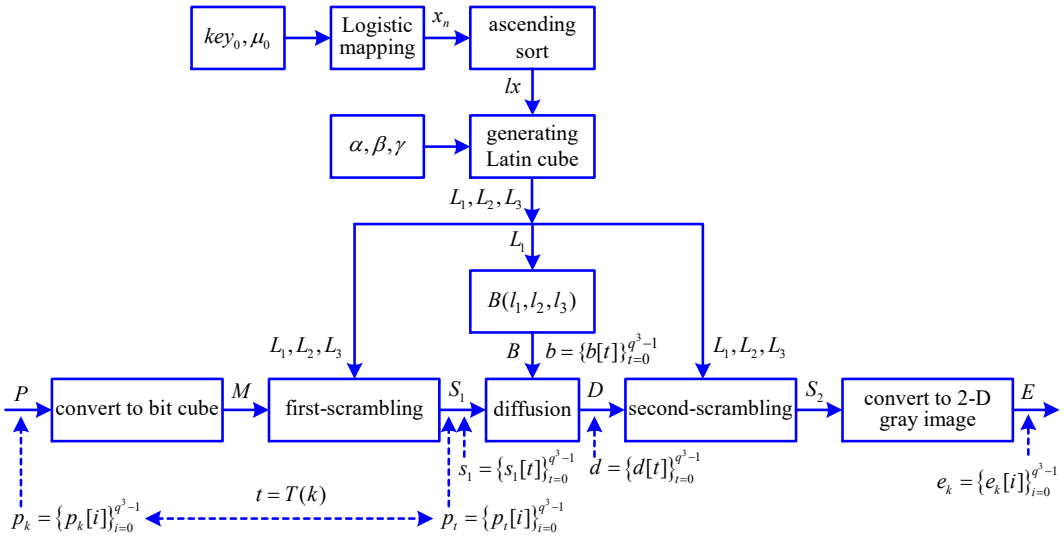

Figure 1. Block diagram of an image chaotic encryption algorithm.

2.2. Logistic Map

According to Figure 1, the chaotic sequence is generated through logistic mapping, given by

$$x_{n+1} = \mu x_n (1 - x_n), \tag{1}$$

where $n = 0, 1, 2, \cdots$, $x_n \in (0, 1)$, $0 \leq \mu \leq 4$. When $\mu > 3.573815$, Equation (1) is chaotic.

2.3. Generation of Latin Cubes

Let the side length of L_1, L_2, L_3 be $q = \sqrt[3]{8 \times w \times h}$, where q is an even number. For a given (l_1, l_2, l_3), one gets $L_1(l_1, l_2, l_3) = \psi_1$, $L_2(l_1, l_2, l_3) = \psi_2$, $L_3(l_1, l_2, l_3) = \psi_3$, $0 \leq \psi_1, \psi_2, \psi_3 \leq q-1$. If $(l_1, l_2, l_3) \neq (l_1', l_2', l_3')$, $(\psi_1, \psi_2, \psi_3) \neq (\psi_1', \psi_2', \psi_3')$, then L_1, L_2, L_3 are orthogonal to each other [38]. When $q = 3$, one gets three orthogonal Latin cubes, as shown in Figure 2a, and the corresponding triple tuple is shown in Figure 2b, respectively.

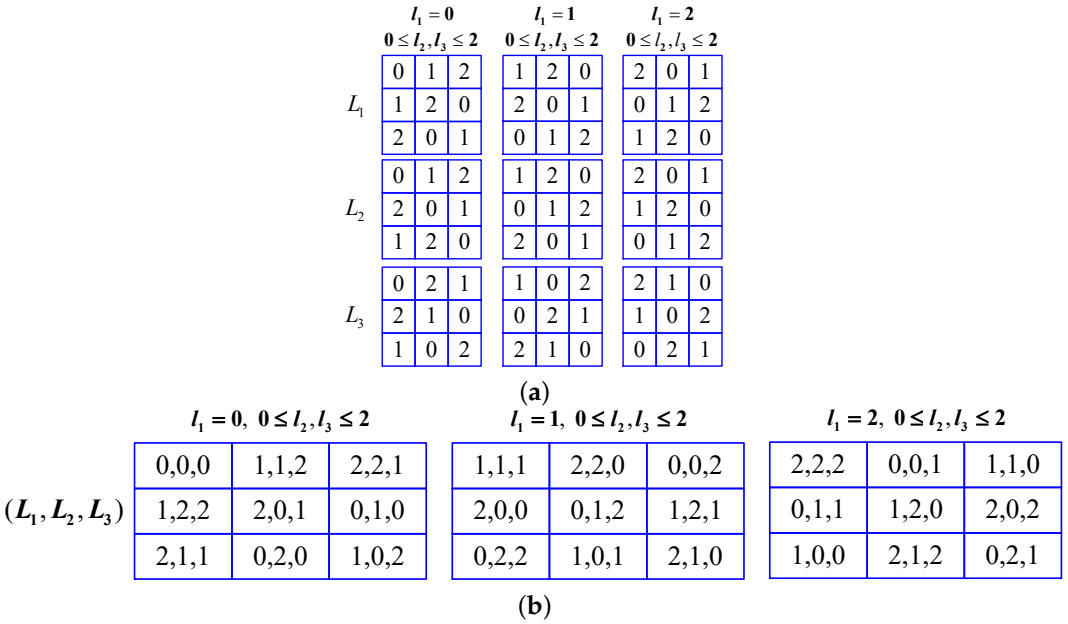

Figure 2. Three orthogonal Latin cubes and the corresponding triple tuple when $q = 3$. (**a**) three orthogonal Latin cubes; (**b**) the corresponding triple tuple.

The algorithm for generating Latin cubes proposed in [20] is implemented by replacing the ordered set $\{0, 1, 2, ..., q\}$ in the generation method proposed in [38] with the chaotic index sequence lx. The detailed steps for generating three orthogonal Latin cubes by means of a finite field are in Algorithm 1.

Algorithm 1 Steps for Generation of Latin Cubes.

Input: Secret keys key_0, μ_0, α, β, γ; Side length $q = \sqrt[3]{8 \times w \times h}$;
Output: Three orthogonal Latin cubes L_1, L_2 and L_3;

1: Generate the chaotic sequence $x = \{x_0, x_1, \ldots, x_{q-1}\}$ by using Logistic mapping.
2: Obtain the corresponding chaotic index sequence $lx = \{c_0, c_1, \cdots, c_i, \cdots, c_{q-1}\}$ by arranging $x = \{x_0, x_1, \ldots, x_{q-1}\}$ in ascending order, where $0 \le c_i,\ i \le q-1$, satisfying $lx[i] = c_i$. Note that the chaotic index sequence lx can only be determined after the sequence value c_i and the sequence number i are simultaneously obtained. When the sequence value c_i is obtained, but the sequence number i is uncertain, the general form of the chaotic index sequence lx is in the form of

$$lx = \{c_{i_0}, c_{i_1}, \cdots, c_{i_k}, \cdots, c_{i_{q-1}}\}, \tag{2}$$

where $0 \le c_{i_k}\ i_k \le q-1$, $i_0 \ne i_1 \ne \cdots \ne i_k \ne \cdots \ne i_{q-1}$, $lx[i_k] = c_{i_k}$. In the following, ξ or ξ' denotes the sequence value and i_ξ or $i'_{\xi'}$ denotes the sequence number in Equation (2), respectively.
3: Construct a finite field by using chaotic index sequence lx, and then one gets the orthogonal Latin cubes on the finite field, given by

$$\begin{cases} L_1(l_1, l_2, l_3) = \alpha^2 \times c_{l_1} + \alpha \times c_{l_2} + c_{l_3}, \\ L_2(l_1, l_2, l_3) = \beta^2 \times c_{l_1} + \beta \times c_{l_2} + c_{l_3}, \\ L_3(l_1, l_2, l_3) = \gamma^2 \times c_{l_1} + \gamma \times c_{l_2} + c_{l_3}, \end{cases} \tag{3}$$

where "$+$" denotes addition operation on the finite field, "$\times$" denotes multiplication operation on the finite field, $\alpha,\ \beta,\ \gamma \in lx$, $c_{l_1}, c_{l_2}, c_{l_3}$ are sequence values of lx.
4: **return** L_1, L_2, L_3.

2.4. Steps for Image Chaotic Encryption

According to Figure 1, and taking a plain gray image with 512×512 resolution as an example, one has $q = \sqrt[3]{512 \times 512 \times 8} = 128$. The steps for image chaotic encryption are in Algorithm 2.

Algorithm 2 Steps for Image Chaotic Encryption.

Input: Secret keys key_0, μ_0, α, β, γ; Plaintext image P;
Output: Ciphertxet image E;

1: Convert the 2D plain gray image P into the bit cube M;
2: Obtain three orthogonal Latin cubes L_1, L_2, L_3 by Algorithm 1;
3: Scramble bit cube M by using three orthogonal Latin cubes L_1, L_2, L_3, and get the corresponding first-scrambling image S_1 in the form of bit cube, such that

$$S_1(l_1, l_2, l_3) = M(L_1(l_1, l_2, l_3), L_2(l_1, l_2, l_3), L_3(l_1, l_2, l_3)). \tag{4}$$

4: Obtain the diffusion bit cube $B(l_1, l_2, l_3)$ by using Latin cube L_1, given by

$$B(l_1, l_2, l_3) = \begin{cases} 0, & \text{if } L_1(l_1, l_2, l_3) \geq 64, \\ 1, & \text{if } L_1(l_1, l_2, l_3) < 64. \end{cases} \tag{5}$$

Then, get the diffusion 1D bit sequence $b[t]$ corresponding to diffusion bit cube $B(l_1, l_2, l_3)$ as

$$b[t] = B\left(\left\lfloor t/128^2 \right\rfloor, \ \lfloor t/128 \rfloor \%128, \ t\%128\right), \tag{6}$$

where $t \in \{0, 1, 2, \cdots, q^3 - 1\}$, $\lfloor \cdot \rfloor$ is a round down operation, and "%" is a modulo operation.
5: Convert $S_1(l_1, l_2, l_3)$ into the 1D bit sequence $s_1[t]$ as

$$s_1[t] = S_1\left(\left\lfloor t/128^2 \right\rfloor, \ \lfloor t/128 \rfloor \%128, \ t\%128\right). \tag{7}$$

Then, get the 1D bit sequence $d[t]$ by using $s_1[t]$ and $b[t]$ as

$$d[t] = s_1[t] \oplus d[t-1] \oplus b[t], \tag{8}$$

where $0 \leq t \leq 128^3 - 1$, $d[-1] = 0$, "$\oplus$" denotes bitwise exclusive or operation.
6: Calculate $G(d) = \left(\sum_{i=0}^{q^3-1} d[i]\right)$, and convert the 1D bit sequence $d[t]$ into the bit cube $D(l_1, l_2, l_3)$. Then, get the bit cube $S_2(l_1, l_2, l_3)$ by utilizing $D(l_1, l_2, l_3)$, such that

$$S_2(l_1, l_2, l_3) = \begin{cases} D(L_2(l_1, l_2, l_3), L_3(l_1, l_2, l_3), L_1(l_1, l_2, l_3)), & (G(d)\%2 = 0), \\ D(L_3(l_1, l_2, l_3), L_1(l_1, l_2, l_3), L_2(l_1, l_2, l_3)), & (G(d)\%2 = 1), \end{cases} \tag{9}$$

where $G(d)\%2 \in \{0, 1\}$ denotes the modular 2 operation on $G(d)$.
7: Convert the bit cube $S_2(l_1, l_2, l_3)$ into the 2D cipher gray image E with 512×512 resolution.
8: **return** E.

An example of encrypting a gray image with 2×4 resolution using the original encryption algorithm is shown in Figure 3. Figure 3a shows the three Latin cubes and the corresponding bit cubes L used for encryption. Figure 3b shows the encryption process. The numbers in the cells of P and E represent pixel values, and the bit values are represented in the cells of S_1, B, and S_2. The red cells in M indicate that they are bit representations of the red cell corresponding to P, i.e., the binary representation of 166 is $(10100110)_2$.

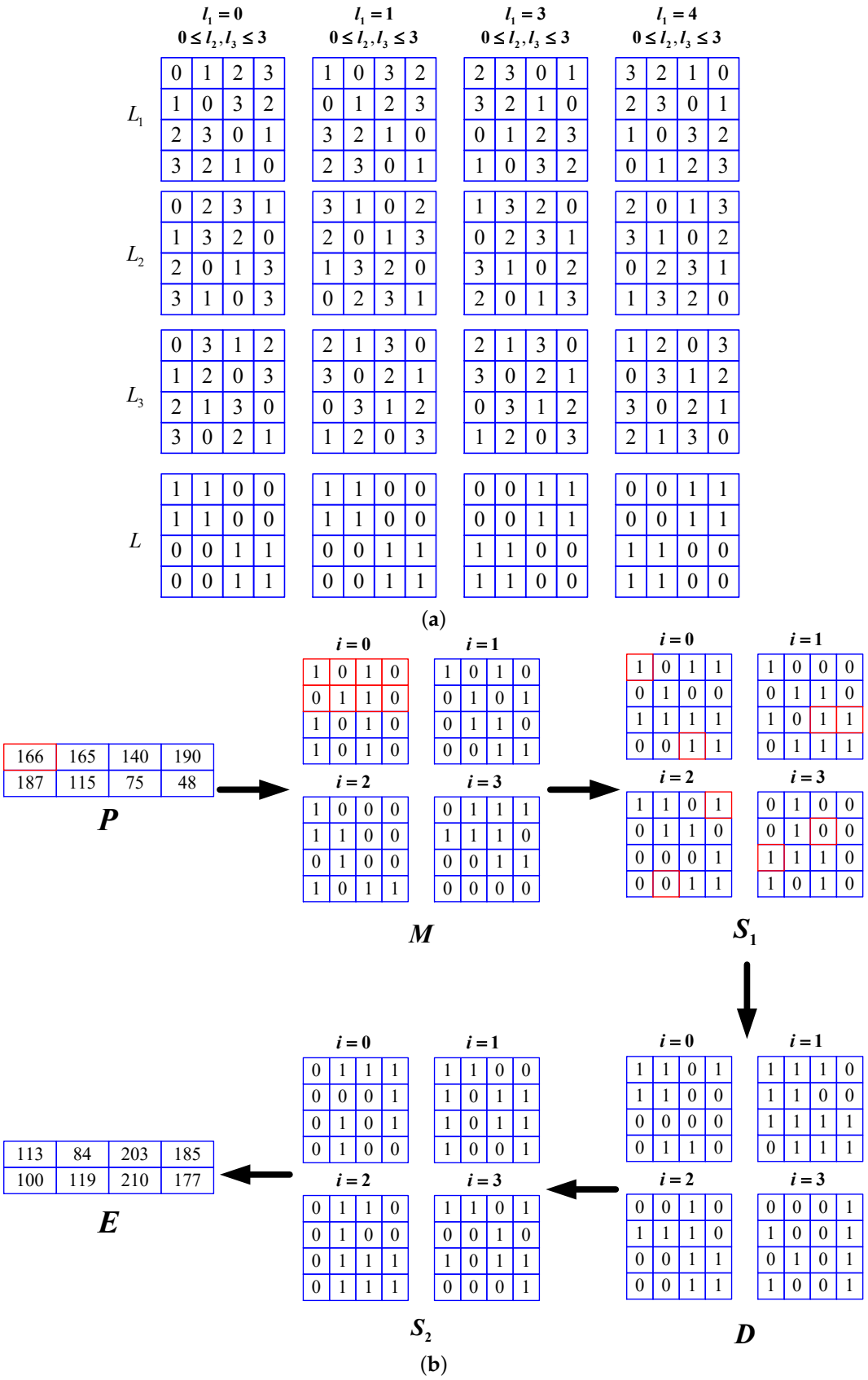

Figure 3. An example of encrypting a gray image with 2×4 resolution. (**a**) three orthogonal Latin cubes and the corresponding bit cubes L used for encryption; (**b**) the encryption process.

3. Security Analysis

According to Figure 1, it is found that the generation of three orthogonal Latin cubes L_1, L_2, L_3 is not related to the plain image. When the secret keys are given, the three orthogonal Latin cubes L_1, L_2, L_3 remain unchanged for different input plain images, which are provided a prerequisite for chosen plaintext attack. Therefore, one can decipher the equivalent secret keys lx, α, β, γ corresponding to the original secret keys key_0, μ_0, α, β, γ.

3.1. Analysis of Chaotic Index Sequence lx

3.1.1. Relation between the First-Scrambling Image S_1 and the Plain Image M

Proposition 1. *Suppose that M is the bit cube representation of P; S_1 is the first-scrambling image of M. The relationship between M and S_1 satisfies $S_1(i_0, i_0, i_\xi) = M(\xi, \xi, \xi)$, where $lx[i_0] = 0$, $lx[i_\xi] = \xi$, i_0, $\xi \in \{0, 1, 2, \cdots, q-1\}$, i_0 denotes the sequence number corresponding to the sequence value 0, and i_ξ denotes the sequence number corresponding to the sequence value ξ.*

Proof. Let $l_1 = l_2 = i_0$, $l_3 = i_\xi$, and substitute them into Equation (4), then, one gets

$$S_1(i_0, i_0, i_\xi) = M(L_1(i_0, i_0, i_\xi), L_2(i_0, i_0, i_\xi), L_3(i_0, i_0, i_\xi)). \tag{10}$$

In addition, let $l_1 = l_2 = i_0$, $l_3 = i_\xi$, and substitute them into Equation (3), then, one gets

$$\begin{cases} L_1(i_0, i_0, i_\xi) = \alpha^2 \times c_{i_0} + \alpha \times c_{i_0} + c_{i_\xi}, \\ L_2(i_0, i_0, i_\xi) = \beta^2 \times c_{i_0} + \beta \times c_{i_0} + c_{i_\xi}, \\ L_3(i_0, i_0, i_\xi) = \gamma^2 \times c_{i_0} + \gamma \times c_{i_0} + c_{i_\xi}. \end{cases} \tag{11}$$

Since $lx[i_0] = 0$, $lx[i_\xi] = \xi$, one has $lx[i_0] = c_{i_0} = 0$, $lx[i_\xi] = c_{i_\xi} = \xi$. In addition, substituting $c_{i_0} = 0$ and $c_{i_\xi} = \xi$ into Equation (11), one gets

$$L_1(i_0, i_0, i_\xi) = L_2(i_0, i_0, i_\xi) = L_3(i_0, i_0, i_\xi) = \xi. \tag{12}$$

In addition, substituting Equation (12) into Equation (10), it follows that $S_1(i_0, i_0, i_\xi) = M(\xi, \xi, \xi)$ holds. The proof is finished. □

3.1.2. The First Case for Analysis of Chaotic Index Sequence lx

Suppose that the 1D bit sequence corresponding to plain image P_0 is $\{p_0[i]\}_{i=0}^{q^3-1} = \{0\}_{i=0}^{q^3-1}$, the cipher image corresponding to plain image P_0 is E_0, the 1D bit sequence corresponding to cipher image E_0 is $\{e_0[i]\}_{i=0}^{q^3-1}$, and the 1D bit sequence corresponding to plain image P_k is $\{p_k[i]\}_{i=0}^{q^3-1}$, where $p_k[i]$ is given by

$$p_k[i] = \begin{cases} 1, & \text{if } i = k, \\ 0, & \text{if } i \neq k. \end{cases} \tag{13}$$

In addition, suppose that the cipher image corresponding to plain image P_k is E_k, the 1D bit sequence corresponding to cipher image E_k is $\{e_k[i]\}_{i=0}^{q^3-1}$, the 1D bit sequence corresponding to plain image $P_{k_1k_2} = P_{k_1} \oplus P_{k_2}$ is $\{p_{k_1k_2}[i]\}_{i=0}^{q^3-1} = \{p_{k_1}[i] \oplus p_{k_2}[i]\}_{i=0}^{q^3-1}$, the cipher image corresponding to plain image $P_{k_1k_2}$ is $E_{k_1k_2}$, the 1D bit sequence corresponding to cipher image $E_{k_1k_2}$ is $\{e_{k_1k_2}[i]\}_{i=0}^{q^3-1}$, the 1D bit sequence corresponding to plain image $P_{k_1k_2k_3} = P_{k_1} \oplus P_{k_2} \oplus P_{k_3}$ is $\{p_{k_1k_2k_3}[i]\}_{i=0}^{q^3-1} = \{p_{k_1}[i] \oplus p_{k_2}[i] \oplus p_{k_3}[i]\}_{i=0}^{q^3-1}$, the cipher image corresponding to $P_{k_1k_2k_3}$ is $E_{k_1k_2k_3}$, and the 1D bit sequence corresponding to $E_{k_1k_2k_3}$ is $\{e_{k_1k_2k_3}[i]\}_{i=0}^{q^3-1}$.

Proposition 2. *Suppose that the cipher image corresponding to plain image P_k is E_k, the 1D bit sequence corresponding to cipher image E_k is $\{e_k[i]\}_{i=0}^{q^3-1}$, the cipher image corresponding to plain image P_0 is E_0, and the 1D bit sequence corresponding to cipher image E_0 is $\{e_0[i]\}_{i=0}^{q^3-1}$. A differential operation is performed in the form of $\sum_{i=0}^{q^3-1}(e_0[i] \oplus e_k[i]) = q^3 - m_{k,0}$, in which $e_k[i_l] = e_0[i_l]$ $(l = 1, 2, \cdots, m_{k,0};\ i_l \in \{0, 1, 2, \cdots, q^3 - 1\})$, $q^3$is an even number. If $(q^3 - m_{k,0})\%2 = q^3\%2 - m_{k,0}\%2 = m_{k,0}\%2 = 0$, then $T(k) = m_{k,0}$ holds, where $T(k)$ denotes the position scrambling rule in the first-scrambling stage, k denotes the position of the k-th bit before the first-scrambling of plain image, and $T(k)$ denotes the position of k-th bit after the first-scrambling of plain image.*

Proof. According to Equation (6), the relationship between the coordinates (l_1, l_2, l_3) of bit cube $B(l_1, l_2, l_3)$ and the position t of 1D bit sequence $b[t]$ corresponding to $B(l_1, l_2, l_3)$ is given by

$$\begin{cases} l_1 = \lfloor t/q^2 \rfloor = \left\lfloor t/128^2 \right\rfloor, \\ l_2 = \lfloor t/q \rfloor \%q = \lfloor t/128 \rfloor \%128, \\ l_3 = t\%q = t\%128. \end{cases} \tag{14}$$

On the other hand, the relationship between the coordinates (ξ, ξ, ξ) of bit cube $M(\xi, \xi, \xi)$ and the position k of 1D bit sequence $p_k[i]$ in Equation (13) is given by

$$k = \xi(q^2 + q + 1). \tag{15}$$

Thus, the relationship between the position of t-th bit after the first-scrambling of plain image and the position of k-th bit before the first-scrambling of plain image is given by

$$t = T(k) = T(\xi(q^2 + q + 1)). \tag{16}$$

(1) Consider the first-scrambling stage. In the first-scrambling stage, only change the bit position, but the bit value should remain unchanged. Suppose that the input 1D bit sequence corresponding to plain image P_k is p_k, after the first-scrambling of plain image, the corresponding output 1D bit sequence is p_t. According to Equation (16), the relationship between position t and k satisfies $t = T(k)$. In particular, if the input 1D bit sequence corresponding to plain image P_0 is $p_0 = \{p_0[i]\}_{i=0}^{q^3-1} = \{0\}_{i=0}^{q^3-1}$, after the first-scrambling of plain image, the corresponding output 1D bit sequence is $p_t = \{p_t[i]\}_{i=0}^{q^3-1}$, then one has $p_t = p_0 = \{0\}_{i=0}^{q^3-1}$. (2) Consider the diffusion stage. Take the output 1D bit encryption sequence $\{p_o[i]\}_{i=0}^{q^3-1}$ in the first-scrambling stage as the input 1D bit sequence in the diffusion stage. According to Equation (8), diffuse $\{p_o[i]\}_{i=0}^{q^3-1}$ by using the diffusion 1D bit sequence $\{b[i]\}_{i=0}^{q^3-1}$, obtain the corresponding output $\{d_o[i]\}_{i=0}^{128^3-1}$ in the diffusion stage. By substituting $s_1[i] = p_o[i] = 0$ into Equation (8), one has

$$\begin{cases} d_o[0] = p_o[0] \oplus d_o[-1] \oplus b[0] = 0 \oplus 0 \oplus b[0] = b[0], \\ d_o[1] = p_o[1] \oplus d_o[0] \oplus b[1] = 0 \oplus d_o[0] \oplus b[1] = b[0] \oplus b[1], \\ d_o[2] = p_o[2] \oplus d_o[1] \oplus b[2] = 0 \oplus d_o[1] \oplus b[2] = b[0] \oplus b[1] \oplus b[2], \\ \quad \cdots \\ d_o[i] = b[0] \oplus b[1] \oplus b[2] \cdots \oplus b\,[i]\,, \end{cases} \tag{17}$$

where $i = 0, 1, 2, \cdots, q^3 - 1$, $d_0[-1] = 0$. Similarly, take the output 1D bit encryption sequence $\{p_t[i]\}_{i=0}^{q^3-1}$ in the first-scrambling stage as the input 1D bit sequence in the diffusion stage. According to Equation (8), diffuse $\{p_t[i]\}_{i=0}^{q^3-1}$ by using the diffusion 1D bit sequence $\{b[i]\}_{i=0}^{q^3-1}$ and obtain the

corresponding output $\{d_t[i]\}_{i=0}^{128^3-1}$ in the diffusion stage. By substituting $s_1[i] = p_t[i]$ into Equation (8), and also by utilizing Equation (17), one has

$$\begin{cases} d_t[0] = p_t[0] \oplus d_t[-1] \oplus b[0] = 0 \oplus 0 \oplus b[0] = d_o[0], \\ d_t[1] = p_t[1] \oplus d_t[0] \oplus b[1] = 0 \oplus d_o[0] \oplus b[1] = 0 \oplus b[0] \oplus b[1] = d_o[1], \\ \cdots \\ d_t[t] = p_t[t] \oplus d_t[t-1] \oplus b[t] = 1 \oplus d_o[t-1] \oplus b[t] = 1 \oplus d_o[t] = \overline{d_o[t]}, \\ d_t[t+1] = p_t[t+1] \oplus d_t[t] \oplus b[t+1] = 0 \oplus d_t[t] \oplus b[t+1] = \overline{d_o[t]} \oplus b[t+1] = \overline{d_o[t+1]}, \\ \cdots \\ d_t[i] = p_t[i] \oplus d_t[i-1] \oplus b[i] = 1 \oplus d_o[i-1] \oplus b[i] = \overline{d_o[i]}, \end{cases} \tag{18}$$

where $d_t[-1] = 0$. According to Equation (18), one has

$$\begin{cases} d_t[i] = d_o[i] & (0 \le i < t), \\ d_t[i] = \overline{d_o[i]} & (t \le i \le (q^3 - 1)), \end{cases} \tag{19}$$

where $\overline{d_o[i]}$ denotes the bitwise NOT of $d_o[i]$. (3) Consider the second-scrambling stage. Take the output 1D bit encryption sequences $\{d_o[i]\}_{i=0}^{q^3-1}$ and $\{d_t[i]\}_{i=0}^{q^3-1}$ in the diffusion stage as the input 1D bit sequences in the second-scrambling stage, calculate $G(d_0) = \left(\sum_{i=0}^{q^3-1} d_0[i]\right)$, $G(d_t) = \left(\sum_{i=0}^{q^3-1} d_t[i]\right)$, respectively. If $t\%2 = 0$ in Equation (19) holds, then it follows that

$$G(d_t)\%2 = G(d_0)\%2. \tag{20}$$

According to Equation (9) with Equation (20), it is noted that the same scrambling rule for $\{d_o[i]\}_{i=0}^{q^3-1}$ and $\{d_t[i]\}_{i=0}^{q^3-1}$ is used in the second-scrambling stage. By comparing the first equation $d_t[i] = d_o[i]$ $(0 \le i < t)$ of Equation (19) with $e_k[i_l] = e_0[i_l]$ $(l = 1, 2, \cdots, m_{k,0};\ i_l \in \{0, 1, 2, \cdots, q^3 - 1\})$, it follows that $t = m_{k,0}$. Then, according to Equation (16), $T(k) = m_{k,0}$ holds. The proof is finished. □

Based on Proposition 1, one has $S_1(i_0, i_0, i_{\xi}) = M(\xi, \xi, \xi)$, where $\xi \in \{0, 1, 2, \cdots, q-1\}$ is the sequence value of chaotic index sequence lx, i_{ξ} is the sequence number of lx. However, even though ξ is given, since $S_1(i_0, i_0, i_{\xi})$ is the first-scrambling result of bit cube $M(\xi, \xi, \xi)$, but the scrambling rule $T(\cdot)$ is unknown beforehand, the sequence numbers i_0 and i_{ξ} cannot be directly available. Thus, Proposition 2 is needed to obtain the specific numbers i_0 and i_{ξ}.

Based on Proposition 2, suppose that the input plain image $M(l_1, l_2, l_3)$ is given by

$$M(l_1, l_2, l_3) = \begin{cases} 1, & \text{if } l_1 = l_2 = l_3 = \xi, \\ 0, & \text{otherwise}, \end{cases} \tag{21}$$

where $\xi \in \{0, 1, \cdots, q-1\}$. Based on Equation (15) with Equation (21), one has $k = \xi \cdot (q^2 + q + 1)$. Next, one obtains $m_{k,0}$ by a chosen plaintext attack. If $m_{k,0}\%2 = 0$ holds, then the same scrambling rule is used for d_0 and d_t in the second-scrambling stage, such that $T(k) = m_{k,0} = t$. Finally, according to Equation (14), it follows that

$$\begin{cases} i_0 = \lfloor t/q^2 \rfloor = \lfloor T(\xi \cdot (q^2 + q + 1))/q^2 \rfloor = \lfloor T(k)/q^2 \rfloor = \lfloor m_{k,0}/q^2 \rfloor, \\ i_{\xi} = t\%q = T(\xi \cdot (q^2 + q + 1))\%q = T(k)\%q = m_{k,0}\%q. \end{cases} \tag{22}$$

An example of Proposition 2 is as in Figure 4. Figure 4a shows the ciphertext corresponding to the grayscale image lena. Figure 4b shows the corresponding ciphertext image after changing the bit at

the bit-cube coordinates (6, 6, 6) of lena. Figure 4c is a bitwise exclusive or result between Figure 4a,b. Figure 4d is a bit statistical histogram of Figure 4c.

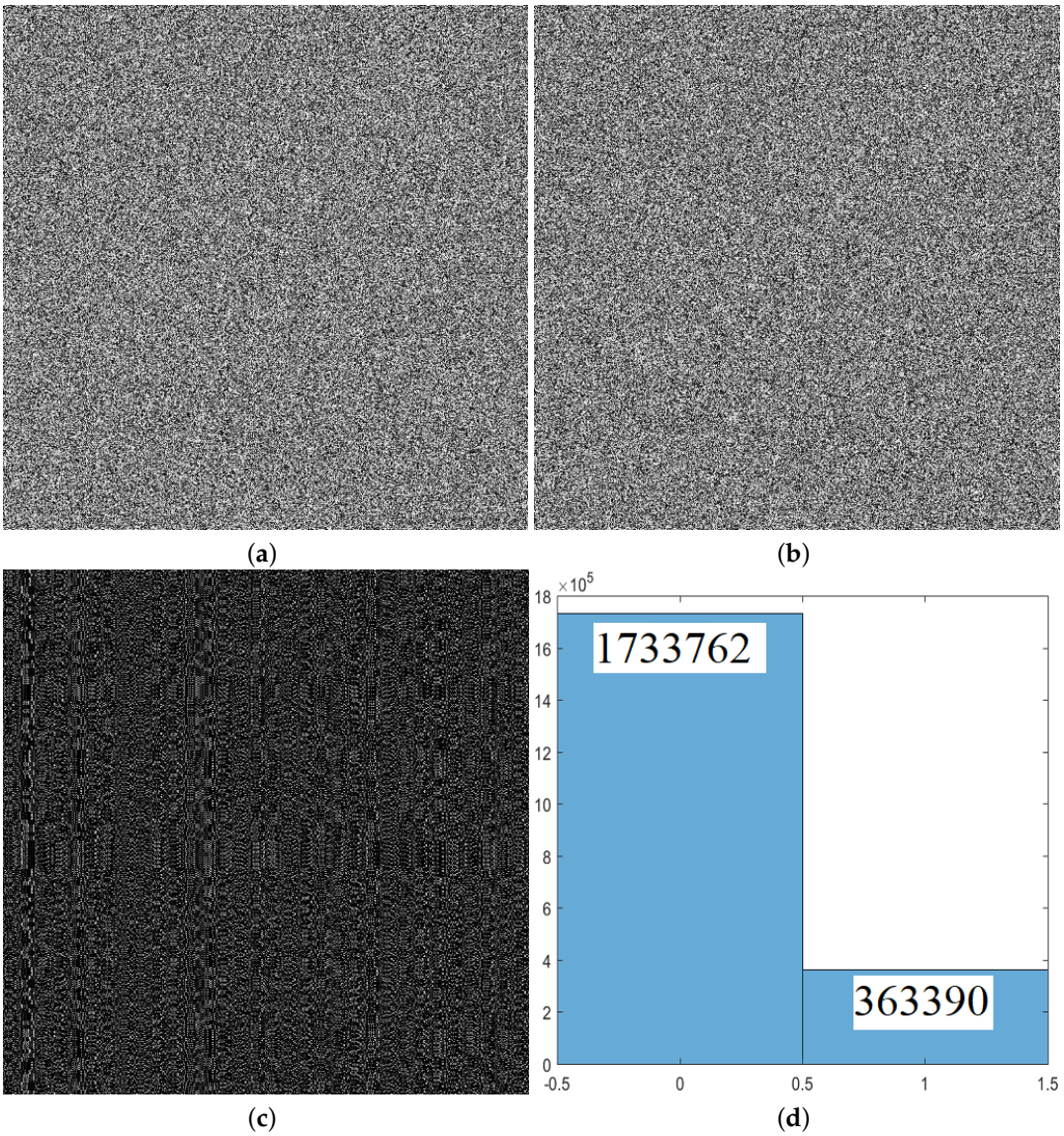

Figure 4. An example of Proposition 2. (**a**) the ciphertext corresponding to the grayscale image lena; (**b**) the corresponding ciphertext image after changing the bit at the bit-cube coordinates (6, 6, 6) of lena; (**c**) the bitwise exclusive or result between Figure 4a,b; (**d**) the bit statistical histogram of Figure 4c.

The difference between the two plaintexts is only 1 bit. It can be found from Figure 4d that the number of identical bits between their corresponding ciphertexts is 1,733,762, which is an even number. Substituting $m_{k,0} = 1,733,762$, $\xi = 6$, and $q = 128$ into Equation (22) yields $i_0 = 105$ and $i_6 = 2$.

3.1.3. The Second Case for Analysis of Chaotic Index Sequence lx

If $m_{k,0}\%2 \neq 0$, the above-mentioned method is no longer available, which needs to be further consideration.

Corollary 1. *Supposing that the cipher image corresponding to plain image* $P_{k_1k_2} = P_{k_1} \oplus P_{k_2}$ $(k_1 \neq k_2)$ *is* $E_{k_1k_2}$*, the 1D bit sequence corresponding to* $E_{k_1k_2}$ *is* $\{e_{k_1k_2}[i]\}_{i=0}^{q^3-1}$*, the cipher image corresponding to plain image* P_0 *is* E_0*, the 1D bit sequence corresponding to* E_0 *is* $\{e_0[i]\}_{i=0}^{q^3-1}$*. A differential operation is performed in the form*

of $\sum_{i=0}^{q^3-1}\left(e_0[i]\oplus e_{k_1k_2}[i]\right)=m_{k_1k_2,0}$, *in which* $e_k[i_l]\neq e_0[i_l]$ $(l=1,2,\cdots,m_{k_1k_2,0};\ i_l\in\{0,1,2,\cdots,q^3-1\})$. *If* $m_{k_1k_2,0}\%2=0$, *then* $|T(k_1)-T(k_2)|=m_{k_1k_2,0}$ *holds. In addition, if* $|T(k_1)-T(k_2)|\%2=0$, *then* $m_{k_1k_2,0}=|T(k_1)-T(k_2)|$ *also holds.*

Corollary 2. *Suppose that the cipher image corresponding to plain image* $P_{k_1k_2k_3}=P_{k_1}\oplus P_{k_2}\oplus P_{k_3}$ $(k_1\neq k_2\neq k_3)$ *is* $E_{k_1k_2k_3}$, *the 1D bit sequence corresponding to* $E_{k_1k_2k_3}$ *is* $\{e_{k_1k_2k_3}[i]\}_{i=0}^{q^3-1}$, *the cipher image corresponding to plain image* P_0 *is* E_0, *the 1D bit sequence corresponding to* E_0 *is* $\{e_0[i]\}_{i=0}^{q^3-1}$. *A differential operation is performed in the form of* $\sum_{i=0}^{q^3-1}\left(e_0[i]\oplus e_{k_1k_2k_3}[i]\right)=q^3-m_{k_1k_2k_3,0}$, *in which* $e_{k_1k_2k_3}[i_l]=e_0[i_l]$ $(l=1,2,\cdots,m_{k_1k_2k_3,0};\ i_l\in\{0,1,2,\cdots,q^3-1\})$, q^3 *is an even number. If* $(q^3-m_{k_1k_2k_3,0})\%2=q^3\%2-m_{k_1k_2k_3,0}\%2=m_{k_1k_2k_3,0}\%2=0$, *then* $T(k_1)+T(k_2)-T(k_3)=m_{k_1k_2k_3,0}$ *holds, where* $T(k_1)<T(k_3)<T(k_2)$ *or* $T(k_1)>T(k_3)>T(k_2)$. *In addition, if* $[T(k_1)+T(k_2)-T(k_3)]\%2=0$, *then* $m_{k_1k_2k_3,0}=T(k_1)+T(k_2)-T(k_3)$ *also holds.*

Suppose that the set of all sequence values corresponding to the chaotic index sequence lx is $\Omega=\{\xi_{i_1},\xi_{i_2},\cdots,\xi_{i_{q/2}},\xi'_{i'_1},\xi'_{i'_2},\cdots,\xi'_{i'_{q/2}}\}$. Let $\Psi=\{\xi_{i_1},\xi_{i_2},\cdots,\xi_{i_{q/2}}\}$ be the set of sequence value ξ corresponding to sequence number i_ξ, where i_ξ is obtained by using Equation (22). The relationship among ξ, k, t is $k=\xi(q^2+q+1)$ and $t=T(k)=T(\xi\cdot(q^2+q+1))$. For $\forall\xi\in\Psi$, $m_{k,0}\%2=0$ and $t=m_{k,0}$ hold. Similarly, let $\Psi'=\{\xi'_{i'_1},\xi'_{i'_2},\cdots,\xi'_{i'_{q/2}}\}$ be the set of sequence value ξ' corresponding to sequence number i'_ξ. The relationship among ξ', k', t' is $k'=\xi'\cdot(q^2+q+1)$ and $t'=T(k')=T(\xi'\cdot(q^2+q+1))$. For $\forall\xi'\in\Psi'$, $m_{k',0}\%2=0$ and $t'=m_{k',0}$ do not hold.

When $\xi\in\Psi$, one has $k=\xi(q^2+q+1)$ and $m_{k,0}\%2=0$, based on the Proposition 2, $t=m_{k,0}$ holds. According to Equation (22), the sequence number i_ξ corresponding to sequence value ξ is given by $i_\xi=t\%q$. However, when $\xi'\in\Psi'$, one has $k'=\xi'(q^2+q+1)$ and $m_{k',0}\%2\neq0$, the Proposition 2 is not available, $t'=m_{k',0}$ does not hold. Therefore, the sequence number $i'_{\xi'}$ corresponding to sequence value $\xi'\in\Psi'$ cannot be determined by using Equation (22).

To further solve the above-mentioned problem, by selecting k'_1, k'_2 $(k'_1\neq k'_2)$, one can obtain $m_{k'_1,0}$ corresponding to k'_1, and $m_{k'_2,0}$ corresponding to k'_2 by using chosen plaintext attack, which satisfies $m_{k'_1,0}\%2=1$ and $m_{k'_2,0}\%2=1$. Under this circumstance, although $T(k'_1)$ and $T(k'_2)$ are unknown, but according to the Proposition 2, $\forall k$ corresponding to $T(k)\%2=0$ can be found, so that the remained $\forall k'$ satisfies $T(k'_1)\%2=1$ and $T(k'_2)\%2=1$, $|T(k'_1)-T(k'_2)|\%2=0$. According to the Corollary 1, it follows that

$$m_{k'_1k'_2,0}=|T(k'_1)-T(k'_2)|=|t'_1-t'_2|. \tag{23}$$

According to the chosen plaintext attack, $m_{k'_1k'_2,0}$ in Equation (23) can be obtained from the given ξ'_1, $\xi'_2\in\Psi'$, where ξ'_1 corresponding to t'_1 satisfies $t'_1=T(\xi'_1(q^2+q+1))$, and ξ'_2 corresponding to t'_2 satisfies $t'_2=T(\xi'_2(q^2+q+1))$, respectively.

For the same k'_1, k'_2, by selecting a suitable k such that $k=\xi(q^2+q+1)$, $m_{k,0}\%2=0$, one gets $\left[T(k'_1)+T(k'_2)-T(k)\right]\%2=0$. Then, according to the Corollary 2, it follows that

$$m_{k'_1k'_2k,0}=T(k'_1)+T(k'_2)-T(k)=t'_1+t'_2-t, \tag{24}$$

where $T(k'_1)<T(k)<T(k'_2)$ or $T(k'_1)>T(k)>T(k'_2)$, $t'_1<t<t'_2$ or $t'_1>t>t'_2$.

According to the chosen plaintext attack, $m_{k'_1k'_2k,0}$ in Equation (24) can be obtained from the given ξ'_1, $\xi'_2\in\Psi'$ and $\xi\in\Psi$, where ξ'_1 corresponding to t'_1 satisfies $t'_1=T(\xi'_1(q^2+q+1))$, ξ'_2 corresponding to t'_2 satisfies $t'_2=T(\xi'_2(q^2+q+1))$, ξ corresponding to t satisfies $t=T(\xi(q^2+q+1))=m_{k,0}$, in which $m_{k,0}$ is known by a chosen plaintext attack as well.

Note that one can also select t'_i,t'_{i+1},t $(i=2,3,\cdots,(q/2-1))$ in the same way, which is omitted here due to the limited length of the article.

According to Equations (23) and (24), four cases are given as follows:

(1) If $t'_1 < t < t'_2$, then one has

$$\begin{cases} t'_2 = (m_{k'_1k'_2,0} + m_{k'_1k'_2k,0} + t)/2 = A_1, \\ t'_1 = (-m_{k'_1k'_2,0} + m_{k'_1k'_2k,0} + t)/2 = B_1. \end{cases} \tag{25}$$

(2) If $t'_1 > t > t'_2$, then one has

$$\begin{cases} t'_1 = (m_{k'_1k'_2,0} + m_{k'_1k'_2k,0} + t)/2 = A_1, \\ t'_2 = (-m_{k'_1k'_2,0} + m_{k'_1k'_2k,0} + t)/2 = B_1. \end{cases} \tag{26}$$

(3) If $t'_2 < t < t'_3$, then one has

$$\begin{cases} t'_3 = (m_{k'_2k'_3,0} + m_{k'_2k'_3k,0} + t)/2 = A_2, \\ t'_2 = (-m_{k'_2k'_3,0} + m_{k'_2k'_3k,0} + t)/2 = B_2. \end{cases} \tag{27}$$

(4) If $t'_2 > t > t'_3$, then one has

$$\begin{cases} t'_2 = (m_{k'_2k'_3,0} + m_{k'_2k'_3k,0} + t)/2 = A_2, \\ t'_3 = (-m_{k'_2k'_3,0} + m_{k'_2k'_3k,0} + t)/2 = B_2. \end{cases} \tag{28}$$

Based on Equations (25)–(28), it follows that

$$\begin{cases} \{t'_1, t'_2\} = \{A_1, B_1\}, \\ \{t'_2, t'_3\} = \{A_2, B_2\}. \end{cases} \tag{29}$$

Then, according to Equation (29), it follows that

$$\begin{cases} t'_2 = \{A_1, B_1\} \bigcap \{A_2, B_2\}, \\ t'_1 = \{A_1, B_1\} - \{t'_2\}, \\ t'_3 = \{A_2, B_2\} - \{t'_2\}. \end{cases} \tag{30}$$

Similarly, for t'_{i-1}, t'_i, t and t'_i, t'_{i+1}, t, one has

$$\begin{cases} t'_i = \{A_{i-1}, B_{i-1}\} \bigcap \{A_i, B_i\}, \\ t'_{i-1} = \{A_{i-1}, B_{i-1}\} - \{t'_i\}, \\ t'_{i+1} = \{A_i, B_i\} - \{t'_i\}, \end{cases} \tag{31}$$

where $i = 2, 3, \cdots, (q/2 - 1)$.

For any given $\xi'_l \in \Psi'$ and $\xi \in \Psi$, according to Equation (31), first, one can get the corresponding t'_l. Then, the sequence number $i'_{\xi'_l}$ corresponding to the sequence value ξ'_l can be further obtained by using t'_l, such that

$$\begin{cases} i'_{\xi'_l} = t'_l \% q, \\ lx[i'_{\xi'_l}] = \xi'_l, \end{cases} \tag{32}$$

where $l \in \{1, 2, \cdots, q/2\}$.

Finally, according to the Equations (22) and (32), one can determine all the sequence values $\xi \in \Psi$, $\xi'_l \in \Psi'$ and all the corresponding sequence numbers i_ξ, $i'_{\xi'}$ in Equation (2), so that the chaotic index sequence lx can be completely deciphered.

3.2. Analysis of Secret Keys α, β, γ

Proposition 3. *Under the condition that the chaotic index sequence lx is obtained, for any $(l_1, l_2, l_3) \neq (l_1', l_2', l_3')$, where $l_i, l_i' \in \{0, 1, 2, \cdots, q-1\}$ $(i = 1, 2, 3)$, if $L_1(l_1, l_2, l_3) = L_2(l_1, l_2, l_3) \neq 0$ and $L_2(l_1', l_2', l_3') = L_3(l_1', l_2', l_3') \neq 0$, then the secret keys α, β, γ can be uniquely determined.*

Proof. According to Equation (3), if $L_1(l_1, l_2, l_3) = L_2(l_1, l_2, l_3) \neq 0$ and $L_2(l_1', l_2', l_3') = L_3(l_1', l_2', l_3') \neq 0$ for any $(l_1, l_2, l_3) \neq (l_1', l_2', l_3')$, then it follows that

$$\begin{cases} L_1(l_1, l_2, l_3) = L_2(l_1, l_2, l_3) = c_{l_1} \times {\chi_1}^2 + c_{l_2} \times \chi_1 + c_{l_3} \neq 0, \\ L_2(l_1', l_2', l_3') = L_3(l_1', l_2', l_3') = c_{l_1'} \times {\chi_2}^2 + c_{l_2'} \times \chi_2 + c_{l_3'} \neq 0, \end{cases} \tag{33}$$

where $c_{l_1}, c_{l_2}, c_{l_3}$ are sequence values of chaotic index sequence lx, $\chi_1 \in \{\alpha, \beta\}$, $\chi_2 \in \{\beta, \gamma\}$.

According to the first equation of Equation (33), one gets two solutions $\chi_1^{(1)}, \chi_1^{(2)}$ for χ_1. Similarly, according to the second equation of Equation (33), one gets two solutions $\chi_2^{(1)}, \chi_2^{(2)}$ for χ_2. Thus, there exists an intersection for the first equation and the second equation of Equation (33), given by $\beta = \{\chi_1^{(1)}, \chi_1^{(2)}\} \bigcap \{\chi_2^{(1)}, \chi_2^{(2)}\}$. Based on the deciphered secret key β, the remaining two secret keys $\alpha = \{\chi_1^{(1)}, \chi_1^{(2)}\} - \{\beta\}$ and $\gamma = \{\chi_2^{(1)}, \chi_2^{(2)}\} - \{\beta\}$ can further be deciphered as well.

If $L_1(l_1, l_2, l_3) = L_2(l_1, l_2, l_3) = 0$ and $L_2(l_1', l_2', l_3') = L_3(l_1', l_2', l_3') = 0$, then, an intersection for the first equation and the second equation of Equation (33) does not exist, so the secret keys α, β, γ cannot be obtained [39]. The proof is finished. □

3.3. Flowchart of Security Analysis

The flowchart of security analysis is shown in Figure 5.

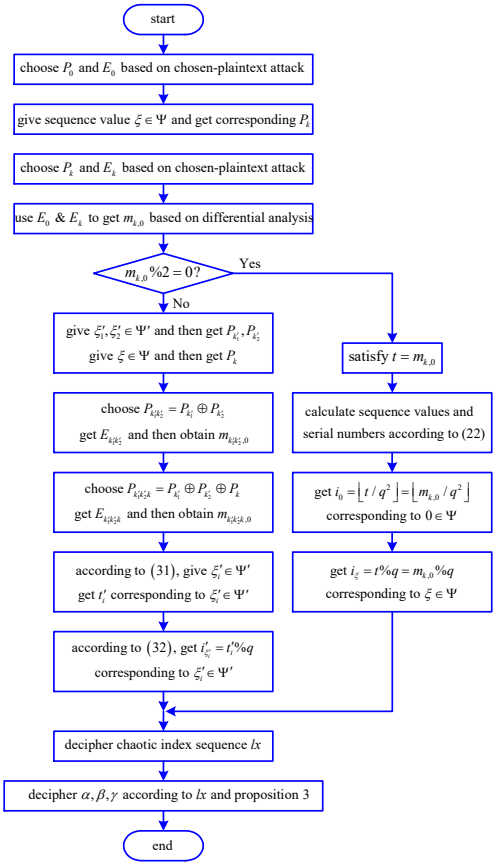

Figure 5. Flowchart of security analysis.

4. Steps for Deciphering the Image Chaotic Encryption Algorithm

The steps for deciphering image chaotic encryption algorithm are as Algorithm 3.

Algorithm 3 Steps for Deciphering Image Chaotic Encryption Algorithm.

Output: The equivalent secret keys lx, α, β, γ;

1: According to the chosen plaintext attack, choose the plain image as P_0, the corresponding cipher image is E_0, the 1D bit sequence corresponding to E_0 is $\{e_0[i]\}_{i=0}^{q^3-1}$.
2: According to the chosen plaintext attack, choose the plain image as P_k, the corresponding cipher image is E_k, the 1D bit sequence corresponding to E_k is $\{e_k[i]\}_{i=0}^{q^3-1}$. where $k = \xi \cdot (q^2+q+1)$, $\xi \in \Psi$.
3: According to the differential attack, calculate $m_{k,0}$ by using $\{e_0[i]\}_{i=0}^{q^3-1}$ and $\{e_k[i]\}_{i=0}^{q^3-1}$ obtained in step 1 and step 2.
4: If $m_{k,0}\%2 = 0$, then $t = m_{k,0}$ holds. According to Equation (22), the sequence number corresponding to sequence value 0 is $i_0 = \lfloor t/q^2 \rfloor = \lfloor m_{k,0}/q^2 \rfloor$, the sequence number corresponding to sequence value $\xi \in \Psi$ is $i_\xi = t\%q = m_{k,0}\%q$.
5: If $m_k\%2 = 1$, then $t \neq m_{k,0}$ holds, Equation (22) is not available. According to the chosen plaintext attack, choose the plain image as $P_{k'_1k'_2} = P_{k'_1} \oplus P_{k'_2}$, the corresponding cipher image is $E_{k'_1k'_2}$, the 1D bit sequence corresponding to $E_{k'_1k'_2}$ is $\{e_{k'_1k'_2}[i]\}_{i=0}^{q^3-1}$. In addition, choose the plain image as$P_{k'_1k'_2k} = P_{k'_1} \oplus P_{k'_2} \oplus P_k$, the corresponding cipher image is $E_{k'_1k'_2k}$, the 1D bit sequence corresponding to $E_{k'_1k'_2k}$ is $\{e_{k'_1k'_2k}[i]\}_{i=0}^{q^3-1}$.
6: According to the differential attack, first calculate $m_{k'_1k'_2,0}$ by using $\{e_0[i]\}_{i=0}^{q^3-1}$ and $\{e_{k'_1k'_2}[i]\}_{i=0}^{q^3-1}$ obtained in step 1 and step 5. Then, calculate $m_{k'_1k'_2k,0}$ by using $\{e_0[i]\}_{i=0}^{q^3-1}$ and $\{e_{k'_1k'_2k}[i]\}_{i=0}^{q^3-1}$ obtained in step 1 and step 5.
7: According to Equation (32), calculate the sequence number $i'_{\xi'_i} = t'_i\%q$ corresponding to sequence value $\xi'_i \in \Psi'$.
8: Decipher the chaotic index sequence lx by using Equation (22) and Equation (32). Then, decipher the secret keys α, β, γ according to the Proposition 3.
9: **return** lx, α, β, γ;

Theoretical analysis and experimental results indicate that only a maximum of $2.5 \times \sqrt[3]{w \times h}$ plain images are needed to decipher the chaotic index sequence lx, and only a maximum of six plain images are needed to decipher secret keys α, β, γ. Therefore, only a maximum of $2.5 \times \sqrt[3]{w \times h} + 6$ is needed to crack the cipher image with $w \times h$ resolution.

5. Numerical Simulation Experiments

In the numerical simulation experiments, the secret keys are set as $key_0 = 0.34$, $\mu_0 = 3.9$, $\alpha = 20$, $\beta = 37$, $\gamma = 46$, the image is with 512×512 resolution. According to the steps for deciphering the image chaotic encryption algorithm given in Section 4, the deciphering algorithm of the origin cipher is implemented by the C program language. Simulations are operated under a laptop computer with Intel Core i7-8550U CPU (Santa Clara, CA, USA) 1.80 GHz, 8 GB RAM, the operating system is Microsoft Windows 10 (Redmond, WA, USA). Using the original algorithm to encrypt and use the algorithm proposed in this paper to crack an image with size of 512×512 takes about 0.115 s and 10.702 s, respectively. Since the encryption process of the algorithm is independent of plaintext and ciphertext, the equivalent key obtained by deciphering any ciphertext image can be used to decipher all ciphertext images of the same resolution. Taking the standard 2D plain gray image Lena, Cameraman, Livingroom as three examples, the plain images, the cipher images, and the deciphered images are shown in Figure 6, respectively.

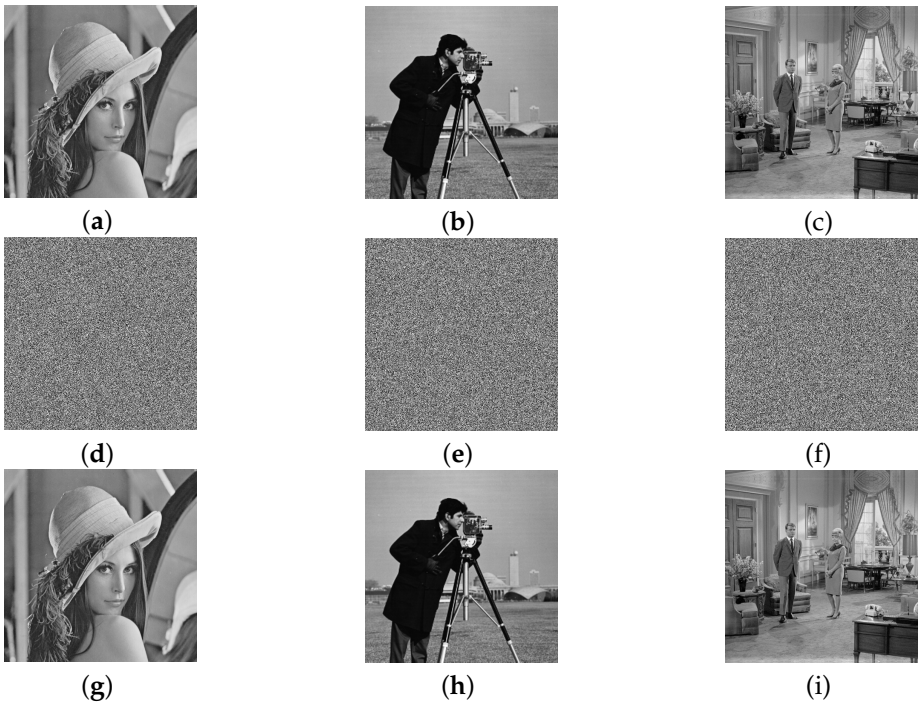

Figure 6. Plain images ((**a**–**c**) row), cipher images ((**d**–**f**) row), and deciphered images ((**g**–**i**) row) of Lena ((**a**–**g**) column), cameraman ((**b**–**h**) column), and living room ((**c**–**i**) column).

Although the previous analysis is for grayscale images, the original encryption algorithm can be easily extended to encrypt color images by encrypting each of the three channels of the color image as a separate grayscale image. In this case, the attack method proposed in this paper is still valid. Take a real-life image with 1024×2048 resolution as an example. Encrypting this image using the original encryption algorithm, it takes about 0.53 s to encrypt the three color channels with the same key, and it takes about 107.36 s to decipher the corresponding ciphertext using the attack method proposed in this paper. Encrypting three color channels with three different sets of keys takes about 1.42 s, and it takes about 318.45 s to decipher the corresponding ciphertext. The results are shown in Figure 7.

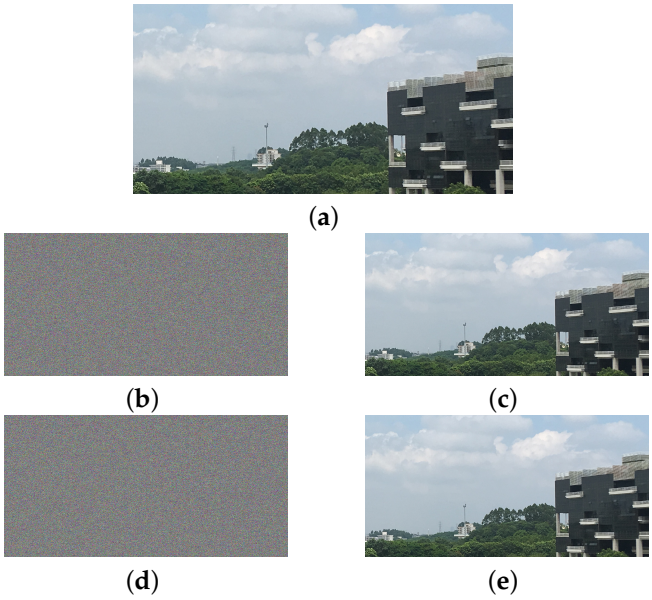

Figure 7. The result of the deciphering of the real-life image. (**a**) the original image; (**b**) encrypting the three color channels with the same key; (**c**) the deciphered image corresponding to (**b**); (**d**) encrypting the three color channels with three different sets of keys; (**e**) the deciphered image corresponding to (**d**).

6. Suggestions for Improvement

According to the analysis in Section 3, the original algorithm is insecure and cannot resist the choice of plaintext attack, and the complexity of the attack method is relatively low. To deal with its security defects, the corresponding suggestions for improvement to enhance the security are as follows:

(1) Enhance the sensitivity of the encryption algorithm to plaintext and ciphertext. According to the analysis in Section 3, the original algorithm has a universal equivalent key lx, α, β, γ. The original algorithm is not sensitive to both plaintext and ciphertext. The root cause of this defect is that the generation of Latin cubes is independent of plaintext image. This vulnerability can be solved by introducing some statistical properties of plaintext, such as the sum of all pixel values, into the generation phase of the Latin cubes.

(2) The mechanism used in the diffusion phase is too simple to achieve the avalanche effect of cryptography, which makes the encryption algorithm vulnerable to differential attacks. To fulfill this demand, increasing the number of encryption rounds or exploiting some complex diffusion mechanisms are worthy options.

7. Conclusions

This paper investigates the security of a Latin-bit cube-based image chaotic encryption algorithm. The algorithm adopts a first-scrambling-diffusion-second-scrambling three-stage encryption scheme. Although the designer claims that the algorithm has passed various statistical tests, the security analysis results in this paper demonstrate that the algorithm has some security vulnerabilities. In particular, the generation of Latin cubes is independent of plain image, and the change in the number of bits in the cipher image follows the change of any one bit in the plain image with obvious regularity. Thus, the equivalent secret keys lx, α, β, γ can be cracked by a chosen plaintext attack and differential attack. Only a maximum of $2.5 \times \sqrt[3]{w \times h} + 6$ plain images are needed to decipher the equivalent secret keys. Theoretical analysis and numerical simulation experiment results verify the effectiveness of the analytical method.

Author Contributions: Methodology, Z.Z.; Project administration, S.Y.; Software, Z.Z.; Supervision, S.Y.; Validation, S.Y.

Funding: This research was funded by the National Key Research and Development Program of China (No. 2016YFB0800401) and the National Natural Science Foundation of China (No. 61532020, 61671161).

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

References

1. Özkaynak, F. Brief review on application of nonlinear dynamics in image encryption. *Nonlinear Dyn.* **2018**, *92*, 305–313. [CrossRef]
2. Abdallah, E.E.; Ben Hamza, A.; Bhattacharya, P. Video watermarking using wavelet transform and tensor algebra. *Signal Image Video Process.* **2010**, *4*, 233–245. [CrossRef]
3. Abdallah, E.E.; Hamza, A.B.; Bhattacharya, P. MPEG Video Watermarking Using Tensor Singular Value Decomposition. In *Image Analysis and Recognition*; Kamel, M., Campilho, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 772–783.
4. Wang, J.; Ding, Q. Dynamic Rounds Chaotic Block Cipher Based on Keyword Abstract Extraction. *Entropy* **2018**, *20*, 693. [CrossRef]
5. Wang, X.; Yang, L.; Liu, R.; Kadir, A. A chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn.* **2010**, *62*, 615–621. [CrossRef]
6. Zhang, Y.; Wang, X. A new image encryption algorithm based on non-adjacent coupled map lattices. *Appl. Soft. Comput.* **2015**, *26*, 10–20. [CrossRef]
7. Wang, X.; Liu, L.; Zhang, Y. A novel chaotic block image encryption algorithm based on dynamic random growth technique. *Opt. Lasers Eng.* **2015**, *66*, 10–18. [CrossRef]

8. Liu, H.; Wang, X. Color image encryption based on one-time keys and robust chaotic maps. *Comput. Math. Appl.* **2010**, *59*, 3320–3327. [CrossRef]
9. Liu, H.; Wang, X. Color image encryption using spatial bit-level permutation and high-dimension chaotic system. *Opt. Commun.* **2011**, *284*, 3895–3903. [CrossRef]
10. Song, C.; Qiao, Y. A Novel Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos. *Entropy* **2015**, *17*, 6954–6968. [CrossRef]
11. Chai, X.; Fu, X.; Gan, Z.; Lu, Y.; Chen, Y. A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process.* **2019**, *155*, 44–62. [CrossRef]
12. Liu, H.; Wang, X.; kadir, A. Image encryption using DNA complementary rule and chaotic maps. *Appl. Soft. Comput.* **2012**, *12*, 1457–1466. [CrossRef]
13. Wang, X.; Zhang, Y.; Bao, X. A novel chaotic image encryption scheme using DNA sequence operations. *Opt. Lasers Eng.* **2015**, *73*, 53–61. [CrossRef]
14. Wang, X.; Feng, L.; Zhao, H. Fast image encryption algorithm based on parallel computing system. *Inf. Sci.* **2019**, *486*, 340–358. [CrossRef]
15. Wang, X.; Gao, S. Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. *Inf. Sci.* **2019**, *507*, 16–36. [CrossRef]
16. Yaghouti Niyat, A.; Moattar, M.H.; Niazi Torshiz, M. Color image encryption based on hybrid hyper-chaotic system and cellular automata. *Opt. Lasers Eng.* **2017**, *90*, 225–237. [CrossRef]
17. Chai, X.; Gan, Z.; Yang, K.; Chen, Y.; Liu, X. An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations. *Signal Process.-Image Commun.* **2017**, *52*, 6–19. [CrossRef]
18. Wang, X.; Li, Z. A color image encryption algorithm based on Hopfield chaotic neural network. *Opt. Lasers Eng.* **2019**, *115*, 107–118. [CrossRef]
19. Bigdeli, N.; Farid, Y.; Afshar, K. A robust hybrid method for image encryption based on Hopfield neural network. *Comput. Electr. Eng.* **2012**, *38*, 356–369. [CrossRef]
20. Xu, M.; Tian, Z. A novel image cipher based on 3D bit matrix and latin cubes. *Inf. Sci.* **2019**, *478*, 1–14. [CrossRef]
21. Xu, M.; Tian, Z. A novel image encryption algorithm based on self-orthogonal Latin squares. *Optik* **2018**, *171*, 891–903. [CrossRef]
22. Chen, J.; Zhu, Z.; Fu, C.; Zhang, L.; Zhang, Y. An efficient image encryption scheme using lookup table-based confusion and diffusion. *Nonlinear Dyn.* **2015**, *81*, 1151–1166. [CrossRef]
23. Zhang, Y.; Wang, X. A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. *Inf. Sci.* **2014**, *273*, 329–351. [CrossRef]
24. Li, M.; Lu, D.; Wen, W.; Ren, H.; Zhang, Y. Cryptanalyzing a Color Image Encryption Scheme Based on Hybrid Hyper-Chaotic System and Cellular Automata. *IEEE Access* **2018**, *6*, 47102–47111. [CrossRef]
25. Wen, H.; Yu, S.; Lü, J. Breaking an Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos. *Entropy* **2019**, *21*, 246. [CrossRef]
26. Li, C.; Lo, K. Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Process.* **2011**, *91*, 949–954. [CrossRef]
27. Jolfaei, A.; Wu, X.; Muthukkumarasamy, V. On the Security of Permutation-Only Image Encryption Schemes. *IEEE Trans. Inf. Forensic Secur.* **2016**, *11*, 235–246. [CrossRef]
28. Feng, W.; He, Y.; Li, H.; Li, C. Cryptanalysis and Improvement of the Image Encryption Scheme Based on 2D Logistic-Adjusted-Sine Map. *IEEE Access* **2019**, *7*, 12584–12597. [CrossRef]
29. Hua, Z.; Zhou, Y. Image encryption using 2D Logistic-adjusted-Sine map. *Inf. Sci.* **2016**, *339*, 237–253. [CrossRef]
30. Lin, Z.; Yu, S.; Lü, J.; Cai, S.; Chen, G. Design and ARM-Embedded Implementation of a Chaotic Map-Based Real-Time Secure Video Communication System. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1203–1216.
31. Lin, Z.; Yu, S.; Feng, X.; Lü, J. Cryptanalysis of a Chaotic Stream Cipher and Its Improved Scheme. *Int. J. Bifurc. Chaos* **2018**, *28*, 1850086. [CrossRef]
32. Hu, G.; Xiao, D.; Wang, Y.; Li, X. Cryptanalysis of a chaotic image cipher using Latin square-based confusion and diffusion. *Nonlinear Dyn.* **2017**, *88*, 1305–1316. [CrossRef]
33. 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]

34. Pak, C. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137. [CrossRef]
35. Wu, J. Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation. *Signal Process.* **2018**, *142*, 292–300. [CrossRef]
36. Zhang, W.; Yu, H.; Zhao, Y.; Zhu, Z. Image encryption based on three-dimensional bit matrix permutation. *Signal Process.* **2016**, *118*, 36–50. [CrossRef]
37. Wang, X.; Teng, L.; Qin, X. A novel colour image encryption algorithm based on chaos. *Signal Process.* **2012**, *92*, 1101–1108. [CrossRef]
38. Arkin, J.; Straus, E.G. Latin k-cubes. *Fibonacci Q.* **1974**, *12*, 288–292.
39. Berlekamp, E.; Rumsey, H.; Solomon, G. On the solution of algebraic equations over finite fields. *Inf. Comput.* **1967**, *10*, 553–564. [CrossRef]

© 2019 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/).