entropy

MDPI

Article

Security Analysis of a Color Image Encryption Algorithm Using a Fractional-Order Chaos

Heping Wen [1,2,*], Chongfu Zhang [1,2], Lan Huang [1], Juxin Ke [3] and Dongqing Xiong [4]

[1] School of Electronic and Information, Zhongshan Institute, University of Electronic Science and Technology of China, Zhongshan 528402, China; cfzhang@uestc.edu.cn (C.Z.); greentree_2001@163.com (L.H.)
[2] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[3] Center of Information and Technology, Dongguan Polytechnic, Dongguan 523808, China; kejx@dgpt.edu.cn
[4] Guangdong Mechanical and Electronical College of Technology, Guangzhou 510515, China; xiongdongqing@gdmec.edu.cn
* Correspondence: hepingwen@yeah.net

Abstract: Fractional-order chaos has complex dynamic behavior characteristics, so its application in secure communication has attracted much attention. Compared with the design of fractional-order chaos-based cipher, there are fewer researches on security analysis. This paper conducts a comprehensive security analysis of a color image encryption algorithm using a fractional-order hyperchaotic system (CIEA-FOHS). Experimental simulation based on excellent numerical statistical results supported that CIEA-FOHS is cryptographically secure. Yet, from the perspective of cryptanalysis, this paper found that CIEA-FOHS can be broken by a chosen-plaintext attack method owing to its some inherent security defects. Firstly, the diffusion part can be eliminated by choosing some special images with all the same pixel values. Secondly, the permutation-only part can be deciphered by some chosen plain images and the corresponding cipher images. Finally, using the equivalent diffusion and permutation keys obtained in the previous two steps, the original plain image can be recovered from a target cipher image. Theoretical analysis and experimental simulations show that the attack method is both effective and efficient. To enhance the security, some suggestions for improvement are given. The reported results would help the designers of chaotic cryptography pay more attention to the gap of complex chaotic system and secure cryptosystem.

Keywords: chaos; image encryption; cryptanalysis

Citation: Wen, H.; Zhang, C.; Huang, L.; Ke, J.; Xiong, D. Security Analysis of a Color Image Encryption Algorithm Using a Fractional-Order Chaos. *Entropy* **2021**, *23*, 258. https://doi.org/10.3390/e23020258

Academic Editor: Amelia Carolina Sparavigna

Received: 1 February 2021
Accepted: 13 February 2021
Published: 23 February 2021

1. Introduction

Nowadays, with the rapid development of optical fiber broadband access network, 5G and other communication technologies, the security of multimedia data, especially digital images, is of particular interest in communication networks [1]. As everyone knows, encryption is an effective means of achieving security enhancements [2]. However, traditional text encryption algorithms such as AES, DES, and IDEA are not suitable for digital images because they featured with strong correlation between adjacent pixels. To deal with the problem, various methodologies are introduced to design different image ciphers. Among them, chaos-based image encryption is the most popular one, because chaos has characteristics of sensitivity to initial values, dense periodic points, and long-term unpredictability of orbits [3–5]. In the past two decades, chaotic image encryption technology has been widely discussed and has become a research hotspot [6]. To improve the security performance of chaotic image encryption technology, various chaotic systems with resistance to dynamic degradation are studied, including quantum chaotic map [7], fractional-order chaos [8], non-degenerated hyperchaos [9], economic chaotic map [10], and cascaded chaotic systems [11], etc. However, chaotic cryptography still lacks authoritative metrics, especially in terms of security. Accordingly, many reported chaotic encryption algorithms have been

broken [12–15]. As shown in Table 1, some previous chaos-based ciphers are vulnerable upon various attack methods, including chosen-ciphertext attack [16], chosen-/known-plaintext attack [12], differential cryptanalysis [17], even cipher-only attack [18]. Therefore, research on security is extremely important and has received much attention [19–33].

Table 1. Some chaos-based ciphers broken by various attack methods.

Ciphers	Broken by	Attack Methods
Fridrich et al. [34] in 1998	Xie et al. [16] in 2017	Chosen-ciphertext attack
Zhao et al. [35] in 2015	Norouzi et al. [36] in 2017	Chosen-plaintext attack
Ye [37] in 2010	Li et al. [18] in 2017	Cipher-only attack
Zhou [38] in 2015	Chen et al. [17] in 2016	Differential cryptanalysis
Song et al. [15] in 2015	Wen et al. [13] in 2019	Chosen-plaintext/cipertext attacks
Shafique et al. [14] in 2018	Wen et al. [12] in 2019	Chosen-plaintext attack

As described in Ref. [39], fractional-order chaotic systems have higher complexity and more optional key parameters and can be used as a competitive encryption scheme. Correspondingly, image encryption algorithms based on fractional-order chaotic systems have attracted the attention of researchers in recent years [35,40–42]. In 2013, Wang et al. [40] introduced a fractional-order chaos into image encryption for the first time, and gave some experiments to verify its performance. Since then, many image encryption schemes based on fractional-order chaotic systems have been proposed [35,41,42]. For example, in 2017, Zhang et al. [41] proposed a color image encryption scheme combing with fractional-order hyperchaotic system and DNA encoding. Yet, cryptanalysts have reported that some fractional-order chaotic image encryption algorithms have some fatal security issues. Exactly, Norouzi et al. [36] pointed out that the image cipher that using an improper fractional-order chaotic system was insecure, which was proposed in [35]. As far as we know, there are still few research studies concerning cryptanalysis on the ciphers based on fractional-order chaotic systems. Moreover, considering that each cryptosystem has its intrinsic characteristics, it is necessary and urgent to perform cryptanalysis on these existing ciphers.

In 2015, a color image encryption algorithm based on a fractional-order hyperchaotic system was proposed [42]. In color image encryption algorithm using a fractional-order hyperchaotic system (CIEA-FOHS), using the pseudo-random sequences generated by the fractional-order hyperchaotic system, RGB-inter permutation, RGB-intra permutation and pixel diffusion are successively performed to get cipher images from plain images. Meanwhile, the relevant pixel correlation, histogram and other experimental analysis are given to verify its security performance. However, from the perspective of cryptanalysis, we found some security defects as follows:

- The existence of an equivalent key. CIEA-FOHS encrypts the image using a pseudo-random sequence generated by fractional-order chaos. However, these sequences are not related to plaintext. Thus, these sequences can be considered as equivalent keys.
- Two-stage permutations can be equivalently simplified to only once. The reason is that the two permutations only change the position of the pixel without changing the value of the pixel.
- The paradigm of the diffusion part is insecure. According to the conclusion of Ref. [43], a class of diffusion encryption using module addition and XOR operations can be cracked with only two special plain images and their corresponding cipher images. Unfortunately, CIEA-FOHS is also the case.

Based on the three points, CIEA-FOHS cannot resist against a chosen-plaintext attack method with the divide-and-conquer strategy. More specifically, under the scenario of chosen-plaintext attack, firstly an equivalent diffusion key is obtained, and then an equivalent permutation key is achieved, and finally the original images can be restored from the encrypted images with the equivalent keys.

2. The Encryption Algorithm under Study

In this section, the fractional-order hyperchaotic system used in Reference [42] is presented, and then the specific steps of CIEA-FOHS are introduced.

2.1. Fractional-Order Hyperchaotic System

The fractional-order hyperchaotic system used in CIEA-FOHS is derived from Ref. [39], given as

$$\begin{cases} D_t^{\alpha} x(t) = -z - w \\ D_t^{\alpha} y(t) = 2y + z \\ D_t^{\alpha} z(t) = 14x - 14y \\ D_t^{\alpha} w(t) = 100(x - g(w)) \end{cases} \tag{1}$$

where x, y, z, w are the four state variables, $g(w) = w - (|w - 0.4| - |w - 0.8| - |w + 0.4| - |w + 0.8|)$, D_t^{α} is the fractional derivative under the definition of Caputo and α is the derivative order. The attractor of the fractional-order hyperchaotic system is shown in Figure 1.

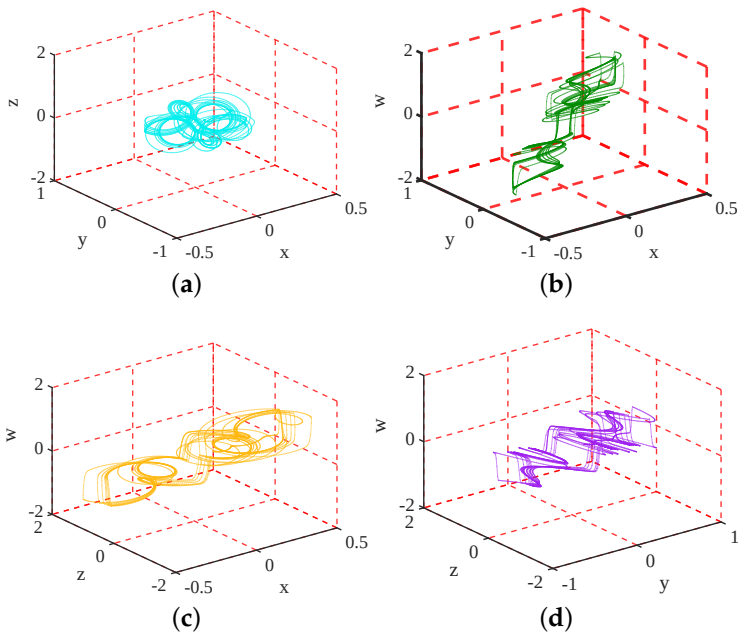


Figure 1. Attractor phase diagrams of the fractional-order hyperchaotic system with different variables: (**a**) (x, y, z); (**b**) (x, y, w); (**c**) (x, z, w); (**d**) (y, z, w).

2.2. Description of CIEA-FOHS

As shown in Figure 2, CIEA-FOHS consists of three main parts: inter-permutation, intra-permutation and pixel diffusion. It is noted that, a two-dimensional image is transformed into an one-dimensional sequence in raster scan order. Specifically, a color plain image $\boldsymbol{I}$ of size $H \times W \times 3$ is converted into three sequences of length $H \times W$ expressed as: $\boldsymbol{IR}$, $\boldsymbol{IG}$, and $\boldsymbol{IB}$, which correspond to the three RGB channels of the image. The main contents are briefly introduced as follows:

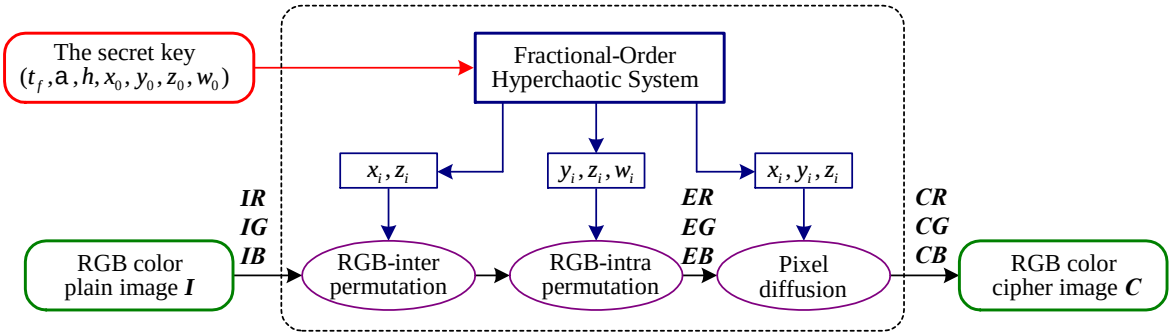


Figure 2. The block diagram of CIEA-FOHS.

- The Secret Key:
  The secret keys of CIEA-FOHS include $(t_f, \alpha, h, x_0, y_0, z_0, w_0)$, where t_f is the fractional derivative defined by Caputo definition, α is the dimension, h is the step size for discretization, and (x_0, y_0, z_0, w_0) are the four initial values of the fractional-order hyperchaotic system defined in Equation (1), respectively. In CIEA-FOHS, these keys are used to generate some chaos-based pseudo-random sequences for encryption [42].
- Initialization:
  In Equation (1), by selecting the secret key as the initial values and parameters and iterating L times, one gets four chaos-based pseudo-random sequences $\{x_i\}_{i=1}^{L}$, $\{y_i\}_{i=1}^{L}$, $\{z_i\}_{i=1}^{L}$ and $\{w_i\}_{i=1}^{L}$, where $L = H \times W$ represents the number of pixels in a single image channel.
- Stage 1. RGB-inter permutation:
  The RGB-inter permutation refers to the process of pixel replacement between channels. This stage is implemented by two control vectors $\{selE_i\}_{i=1}^{L}$ and $\{selLen_i\}_{i=1}^{L}$, which are given as

$$\begin{cases} selE_i = (|x_i| \times 10^{14}) \bmod 6 \\ selLen_i = (|z_i| \times 10^{14}) \bmod 3 \end{cases} \tag{2}$$

  where $i = 1 \sim L$. More specifically, $\{selE_i\}_{i=1}^{L}$ is used to switch channels, as shown in Table 2, and $\{selLen_i\}_{i=1}^{L}$ is to control the position and length of the permutation pixel, given as

Table 2. The stutas of RGB-inter permutation under six rules.

Rule $selE(i)$	0	1	2	3	4	5
Permutation status	$R \to R$ $G \to G$ $B \to B$	$R \to R$ $G \to B$ $B \to G$	$R \to G$ $G \to R$ $B \to B$	$R \to B$ $G \to R$ $B \to G$	$R \to G$ $G \to B$ $B \to R$	$R \to B$ $G \to G$ $B \to R$

$$\begin{cases} length = (sum(ER(pos : pos + length - 1)) \bmod 64), \text{if } selLen_i = 0 \\ length = (sum(EG(pos : pos + length - 1)) \bmod 64), \text{if } selLen_i = 1 \\ length = (sum(EB(pos : pos + length - 1)) \bmod 64), \text{if } selLen_i = 2 \end{cases} \tag{3}$$

  where pos is the starting position, $length$ is the length of the permautation pixels, and sum is the cumulative function.
- Stage 2. RGB-intra permutation:
  Sort $\{y_i\}_{i=1}^{L}$, $\{z_i\}_{i=1}^{L}$, and $\{w_i\}_{i=1}^{L}$ to get three index sequences $\{IY_i\}_{i=1}^{L}$, $\{IZ_i\}_{i=1}^{L}$, and $\{IW_i\}_{i=1}^{L}$ respectively, and their values range $[1, L]$. Use $\{IY_i\}_{i=1}^{L}$, $\{IZ_i\}_{i=1}^{L}$, and

$\{IW_i\}_{i=1}^{L}$ to permute $\boldsymbol{ER}$, $\boldsymbol{EG}$ and $\boldsymbol{EB}$ respectively, given as $ER_i = ER(IY_i)$, $EG_i = EG(IZ_i)$ and $EB_i = EB(IW_i)$.

- Stage 3. Pixel diffusion:
  Perform pixel diffusion on $\boldsymbol{ER}$, $\boldsymbol{EG}$ and $\boldsymbol{EB}$, and then get three channels of the cipher image $\boldsymbol{C}$. Exactly, the three channels $\boldsymbol{CR}$, $\boldsymbol{CG}$ and $\boldsymbol{CB}$ are defined as

$$\begin{cases} CR_i = SX_i \oplus ((ER_i + SX_i) \bmod 256) \oplus CR_{i-1} \\ CG_i = SY_i \oplus ((EG_i + SY_i) \bmod 256) \oplus CG_{i-1} \\ CB_i = SZ_i \oplus ((EB_i + SZ_i) \bmod 256) \oplus CB_{i-1} \end{cases} \tag{4}$$

  where $i = 1 \sim L$, $\oplus$ is bitwise XOR operation, mod represents modulo operation, and $CR_0 = SX_L$, $CG_0 = SY_L$, and $CB_0 = SZ_L$. Here, three diffusion sequences $\boldsymbol{SX}$, $\boldsymbol{SY}$ and $\boldsymbol{SZ}$ are generated by $SX_i = round(x_i) \times 10^{14}$, $SY_i = round(y_i) \times 10^{14}$ and $SZ_i = round(z_i) \times 10^{14}$ respectively, where *round* is a rounding operation on real numbers.

  Decryption is the inverse of encryption and is not described in detail here.

3. Security Analysis of CIEA-FOHS

3.1. Preliminary Analysis of CIEA-FOHS

Referring to the basic assumptions of cryptanalysis, everything about the cryptosystem is public and only the secret key is unknown for attackers [13]. Chosen-plaintext attack is a common and powerful method of cryptanalysis. It assumes that attackers can arbitrarily choose the plaintext that is conducive to deciphering and obtain the corresponding ciphertext [12]. Under the scenario of chosen-plaintext attack, attackers can construct special plain images, such as all black and all white, and obtain the corresponding cipher images to analyze the target cipher.

From the perspective of cryptanalysis, two-stage permutations of CIEA-FOHS can be treated as a global pixel permutation because they only change the pixels' position without their values. The difference is that the number of pixels performing the permutation is $3HW$ instead of HW. Then, the algorithm structure of CIEA-FOHS is actually a classic single-round permutation-diffusion. Moreover, the generation process of all chaos-based pseudo-random sequences is independent of the plain image, which means that these sequences can be regarded as an equivalent key. The reason is that, in the case of a given secret key, these sequences are fixed for encrypting different plain images with the same size. Then, CIEA-FOHS can be equivalently simplified as Figure 3, where $\boldsymbol{PM}$ is an equivalent permutation key and three diffusion sequences $\boldsymbol{SX}$, $\boldsymbol{SY}$ and $\boldsymbol{SZ}$ serve as an equivalent diffusion key.

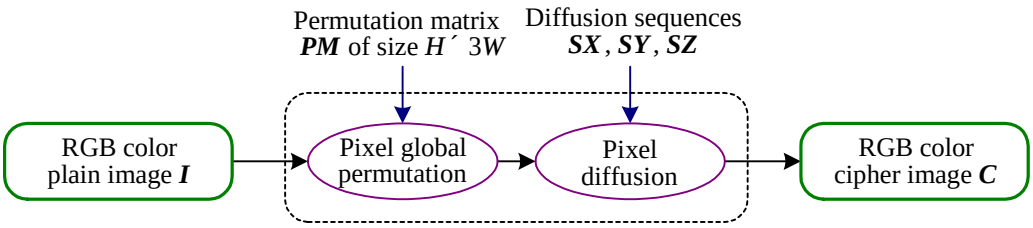


Figure 3. The block diagram of an equivalent simplified CIEA-FOHS.

Based on the above, under the scenario of chosen-plaintext attack and the strategy of divide and conquer, one can get the equivalent keys and then recover the original plain images. Specifically, firstly choose some plain images with same pixel values to cancel the permutation and get the corresponding plain images to obtain the diffusion key; then achieve the permutation key by the method of Reference [12]; finally, recover the images by the equivalent keys.

3.2. Analysis on the Diffusion Part

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

- *Step 1.* Choose the all-zero plain image $\boldsymbol{I}^{(0)}$ and get the corresponding cipher image $\boldsymbol{C}^{(0)}$ to determine SX_L, SY_L, SZ_L.
  The reason for choosing the all-zero image is that the permutation is invalid at this time, and the diffusion can be eliminated to the greatest extent. Then, Equation (4) becomes

$$\begin{cases} CR_i^{(0)} = CR_{i-1}^{(0)} \\ CG_i^{(0)} = CG_{i-1}^{(0)} \\ CB_i^{(0)} = CB_{i-1}^{(0)} \end{cases} \tag{5}$$

  when $i = 1$, one has $CR_1^{(0)} = CR_0$. Since $CR_0 = SX_L$, thus $SX_L = CR_i^{(0)}$. Similarly, one further gets $SY_L = CG_i^{(0)}$ and $SZ_L = CB_i^{(0)}$.
- *Step 2.* Choose two special plain images and get the corresponding cipher images to determine SX_i, SY_i, SZ_i for $i = 1 \sim L-1$.
  Referring to [43,44], the two chosen plaintexts are pure-color images with pixel values of 85 and 170, represented as $\boldsymbol{I}^{(85)}$ and $\boldsymbol{I}^{(170)}$, respectively. Because for the combined operation of module addition and bitwise XOR, choosing these two plain images can minimize the number of solutions for $\boldsymbol{SX}$, $\boldsymbol{SY}$, $\boldsymbol{SZ}$. Under the plain image $\boldsymbol{I}^{(85)}$ and its corresponding cipher image $\boldsymbol{C}^{(85)}$, one gets

$$\begin{cases} CR_i^{(85)} = SX_i \oplus ((85 + SX_i) \bmod 256) \oplus CR_{i-1}^{(85)} \\ CG_i^{(85)} = SY_i \oplus ((85 + SY_i) \bmod 256) \oplus CG_{i-1}^{(85)} \\ CB_i^{(85)} = SZ_i \oplus ((85 + SZ_i) \bmod 256) \oplus CB_{i-1}^{(85)} \end{cases} \tag{6}$$

  Similarly, given the plain image $\boldsymbol{I}^{(170)}$ and its corresponding cipher image $\boldsymbol{C}^{(170)}$, one has

$$\begin{cases} CR_i^{(170)} = SX_i \oplus ((170 + SX_i) \bmod 256) \oplus CR_{i-1}^{(170)} \\ CG_i^{(170)} = SY_i \oplus ((170 + SY_i) \bmod 256) \oplus CG_{i-1}^{(170)} \\ CB_i^{(170)} = SZ_i \oplus ((170 + SZ_i) \bmod 256) \oplus CB_{i-1}^{(170)} \end{cases} \tag{7}$$

  By performing bitwise on Equations (6) and (7), one further gets

$$\begin{cases} (85 \dot{+} SX_i) \oplus (170 \dot{+} SX_i) = CR_i^{(85)} \oplus CR_{i-1}^{(85)} \oplus CR_i^{(170)} \oplus CR_{i-1}^{(170)} \\ (85 \dot{+} SY_i) \oplus (170 \dot{+} SY_i) = CG_i^{(85)} \oplus CG_{i-1}^{(85)} \oplus CG_i^{(170)} \oplus CG_{i-1}^{(170)} \\ (85 \dot{+} SZ_i) \oplus (170 \dot{+} SZ_i) = CB_i^{(85)} \oplus CB_{i-1}^{(85)} \oplus CB_i^{(170)} \oplus CB_{i-1}^{(170)} \end{cases} \tag{8}$$

  where $\dot{+}$ is defined as $a \dot{+} b \triangleq \bmod(a + b, 256)$. It is worth pointing out that the reason why 85 and 170 are chosen as the attack images is that their binary are 01010101 and 10101010 respectively. At this time, the number of possible solutions of SX_i, SY_i, SZ_i is the smallest, which is two. More precisely, the difference between the two solutions is 128. Then, based on Equation (8), we propose Alogrithm 1 to determine SX_i, SY_i, SZ_i, where $i = 1 \sim L-1$.
- *Step 3.* Eliminate the diffusion part by $\boldsymbol{SX}$, $\boldsymbol{SY}$, $\boldsymbol{SZ}$.

Corresponding to Equation (4), the decryption process of diffusion is given as

$$\begin{cases} ER_i = (SX_i \oplus CR_i \oplus CR_{i-1} - SX_i) \bmod 256 \\ EG_i = (SY_i \oplus CG_i \oplus CG_{i-1} - SY_i) \bmod 256 \\ EB_i = (SZ_i \oplus CB_i \oplus CB_{i-1} - SZ_i) \bmod 256 \end{cases} \tag{9}$$

Thus, $\boldsymbol{ER}$, $\boldsymbol{EG}$, $\boldsymbol{EB}$ can be restored from $\boldsymbol{CR}$, $\boldsymbol{CG}$, $\boldsymbol{CB}$ with $\boldsymbol{SX}$, $\boldsymbol{SY}$, $\boldsymbol{SZ}$, respectively.

Algorithm 1: Determining SX_i, SY_i, SZ_i for $i = 1 \sim L-1$

Input: SX_L, SY_L, SZ_L, two chosen plain images $\boldsymbol{I}^{(85)}$ and $\boldsymbol{I}^{(170)}$, and their corresponding cipher images $\boldsymbol{C}^{(85)}$ and $\boldsymbol{C}^{(170)}$.
Output: SX_i, SY_i, SZ_i for $i = 1 \sim L-1$

$i \leftarrow 1$;
for $x \leftarrow 0$ *to* 255 **do**
  if $(85\dot{+}x) \oplus (170\dot{+}x) = CR_1^{(85)} \oplus CR_1^{(170)}$ **then**
    $SX_1 \leftarrow x$;
  end
  if $(85\dot{+}x) \oplus (170\dot{+}x) = CG_1^{(85)} \oplus CG_1^{(170)}$ **then**
    $SY_1 \leftarrow x$;
  end
  if $(85\dot{+}x) \oplus (170\dot{+}x) = CB_1^{(85)} \oplus CB_1^{(170)}$ **then**
    $SZ_1 \leftarrow x$;
  end
end
for $i \leftarrow 2$ *to* $L-1$ **do**
  for $x \leftarrow 0$ *to* 255 **do**
    if $(85\dot{+}x) \oplus (170\dot{+}x) = CR_i^{(85)} \oplus CR_{i-1}^{(85)} \oplus CR_i^{(170)} \oplus CR_{i-1}^{(170)}$ **then**
      $SX_i \leftarrow x$;
    end
    if $(85\dot{+}x) \oplus (170\dot{+}x) = CG_i^{(85)} \oplus CG_{i-1}^{(85)} \oplus CG_i^{(170)} \oplus CG_{i-1}^{(170)}$ **then**
      $SY_i \leftarrow x$;
    end
    if $(85\dot{+}x) \oplus (170\dot{+}x) = CB_i^{(85)} \oplus CB_{i-1}^{(85)} \oplus CB_i^{(170)} \oplus CB_{i-1}^{(170)}$ **then**
      $SZ_i \leftarrow x$;
    end
  end
end
return SX_i, SY_i, SZ_i for $i = 1 \sim L-1$

3.3. Analysis on the Permutation Part

Once the diffusion part is broken, CIEA-FOHS degenerates into a permutation-only cipher. Based on existing research, it cannot resist a chosen-plaintext attack. The basic idea of attacking permutation-only is to construct a special plain image with unequal element values, and get the corresponding permuted image. Taking $2 \times 2 \times 3$ as an example, the process of solving $\boldsymbol{PM}$ is described below. First, a chosen plain image and the corresponding permuted image are given as

$$\boldsymbol{IR} = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}; \boldsymbol{IG} = \begin{bmatrix} 4 & 5 \\ 6 & 7 \end{bmatrix}; \boldsymbol{IB} = \begin{bmatrix} 8 & 9 \\ 10 & 11 \end{bmatrix}$$

$$\boldsymbol{ER} = \begin{bmatrix} 5 & 8 \\ 3 & 11 \end{bmatrix}; \boldsymbol{EG} = \begin{bmatrix} 1 & 10 \\ 2 & 9 \end{bmatrix}; \boldsymbol{EB} = \begin{bmatrix} 6 & 4 \\ 0 & 7 \end{bmatrix}$$

For ease of explanation, a matrix of size $H \times 3W$ is obtained by connecting three channels of size $H \times W$ in a row connection manner. Then, the permutation process can be described by

$$\begin{bmatrix} 0 & 1 & 4 & 5 & 8 & 9 \\ 2 & 3 & 6 & 7 & 10 & 11 \end{bmatrix} \xrightarrow{\boldsymbol{PM}} \begin{bmatrix} 5 & 8 & 1 & 10 & 6 & 4 \\ 3 & 11 & 2 & 9 & 0 & 7 \end{bmatrix}$$

where $\boldsymbol{PM}$ is the permutation matrix of size $H \times 3W$. Finally, $\boldsymbol{PM}$ is determined as

$$\boldsymbol{PM} = \begin{bmatrix} (2,5) & (1,3) & (1,6) & (1,1) & (1,2) & (2,4) \\ (2,3) & (2,1) & (1,5) & (2,6) & (1,4) & (2,2) \end{bmatrix} \tag{10}$$

Obviously, one can recover $(\boldsymbol{IR}, \boldsymbol{IG}, \boldsymbol{IB})$ from $(\boldsymbol{ER}, \boldsymbol{EG}, \boldsymbol{EB})$ with $\boldsymbol{PM}$. However, the situation may be more complicated for large size images. For an 8-bit image, the pixel value range is $[0, 255]$. Thus, when $3HW > 256$, $\boldsymbol{PM}$ cannot be determined by only one chosen plain image and its corresponding cipher image. Fortunately, this problem has been solved in our latest research [12,13]. The basic idea is to combine multiple chosen plain images in a weighted manner to form a matrix with different elements, and the number of chosen plain images required for attacking permutation is $\lceil \log_{256}(3HW) \rceil$, where $\lceil . \rceil$ is the rounding up operation.

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

- *Step 1.* Choose some special plain images and get their corresponding cipher images to determine the permutation matrix $\boldsymbol{PM}$;
- *Step 2.* Use the permutation matrix $\boldsymbol{PM}$ to recover the original images from the permuted images.

3.4. The Proposed Chosen-Plaintext Attack Method

Following the above-mentioned discussion, CIEA-FOHS cannot resist the attack method proposed in this paper. The flowchart of the attack method is shown in Figure 4, and the specific steps based on chosen-plaintext attack are given as: firstly, get an equivalent diffusion key $(\boldsymbol{SX}, \boldsymbol{SY}, \boldsymbol{SZ})$ by the method in Section 3.2; secondly, achieve the permutation matrix $\boldsymbol{PM}$ by the method in Section 3.3; finally, recover the original images with the equivalent keys.

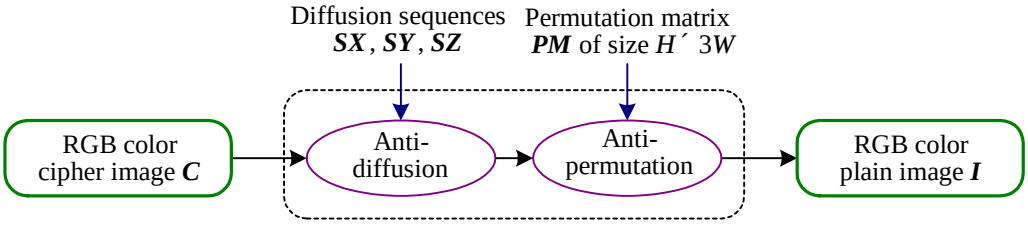


Figure 4. The overall flowchart of attacking CIEA-FOHS.

Moreover, the complexity required for the attack method is discussed here. In terms of data complexity, for color images of size $H \times W \times 3$, the number of chosen plain images required to decipher diffusion and permutation is 3 and $\lceil \log_{256}(3HW) \rceil$, respectively. Hence, the total data complexity required is $\boldsymbol{O}(3 + \lceil \log_{256}(3HW) \rceil)$.

4. Experimental Verifications and Discussions

To verify our security analysis, the algorithm steps of CIEA-FOHS strictly follow Ref. [42]. Although Due to the complexity of fractional-order chaos, some parameters may not be completely consistent, but this does not affect the effectiveness of security analysis. We conduct simulation verification on the proposed image cryptosystem based on a PC (personal computer) with MATLAB r2018b. The running PC is installed with Windows 10 64-bit OS (operating system), Intel(R) Core(TM) i5-8265U CPU @ 1.60 GHz and 8 GB memory. We select some typical images listed in Table 3 for experiments. Among them, the image "Lenna" of size $256 \times 256 \times 3$ given in Ref. [42] is also included. In Equation (1), we set the experimental secret key parameters for $h = 0.001$, $\alpha = 104$, $t_f = 100$, $x_0 = 1.002$, $y_0 = 0.949$, $z_0 = 0.997$ and $w_0 = 1.103$.

- *Case 1.* Breaking CIEA-FOHS with an image of size $2 \times 2 \times 3$:
  In order to better illustrate the attack process, we first adopt an extremely simple image with a size of $2 \times 2 \times 3$. A pair of the given target plain and cipher images $\boldsymbol{I}$ and $\boldsymbol{C}$ is shown in Figure 5a,c respectively, and their histograms are shown in Figure 5b,d respectively. Accordingly, the numerical matrices of $\boldsymbol{I}$ and $\boldsymbol{C}$ are:

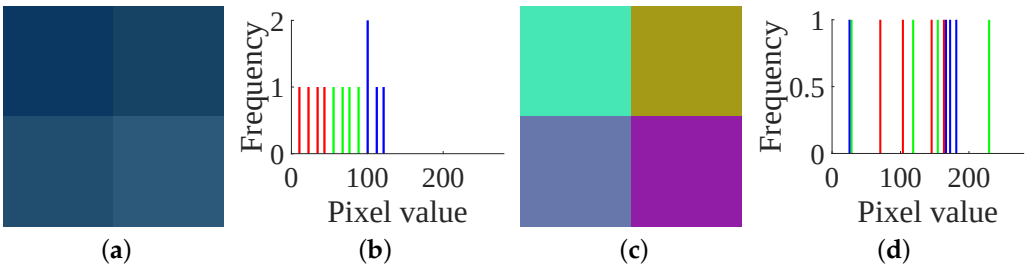


Figure 5. A pair of plain and cipher images of size $2 \times 2 \times 3$: (**a**) plain image $\boldsymbol{I}$; (**b**) histogram of $\boldsymbol{I}$; (**c**) cipher image $\boldsymbol{C}$; (**d**) histogram of $\boldsymbol{C}$.

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

$$\boldsymbol{CR} = \begin{bmatrix} 70 & 165 \\ 103 & 145 \end{bmatrix}; \boldsymbol{CG} = \begin{bmatrix} 231 & 154 \\ 118 & 28 \end{bmatrix}; \boldsymbol{CB} = \begin{bmatrix} 181 & 24 \\ 171 & 165 \end{bmatrix}$$

Firstly, following Step 1 in Section 3.2, choose the all-zero plain image $\boldsymbol{I}^{(0)}$ shown in Figure 6a and temporarily use the encryption machine of CIEA-FOHS, and then get the corresponding cipher image $\boldsymbol{C}^{(0)}$, as shown in Figure 6c. The all-zero plain image $\boldsymbol{I}^{(0)}$ and the corresponding cipher image $\boldsymbol{C}^{(0)}$ and their histograms are shown in Figure 6b,d, respectively. Similarly, the numerical matrices of $\boldsymbol{I}^{(0)}$ and $\boldsymbol{C}^{(0)}$ are:

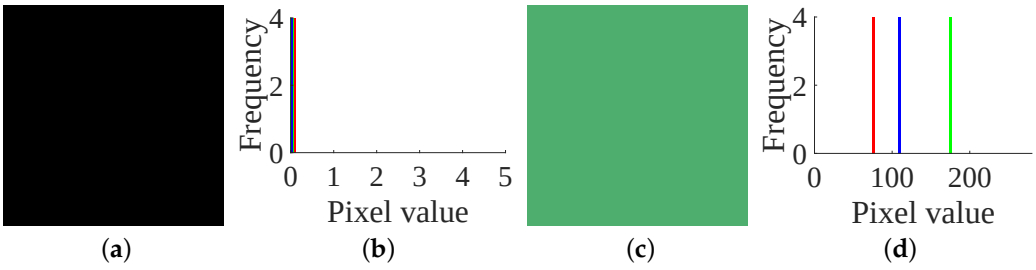


Figure 6. The all-zero chosen plain image $\boldsymbol{I}^{(0)}$ and its corresponding cipher image $\boldsymbol{C}^{(0)}$ of size $2 \times 2 \times 3$: (**a**) $\boldsymbol{I}^{(0)}$; (**b**) histogram of $\boldsymbol{I}^{(0)}$; (**c**) $\boldsymbol{C}^{(0)}$; (**d**) histogram of $\boldsymbol{C}^{(0)}$.

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

$$\boldsymbol{CR}^{(0)} = \begin{bmatrix} 77 & 77 \\ 77 & 77 \end{bmatrix}; \boldsymbol{CG}^{(0)} = \begin{bmatrix} 174 & 174 \\ 174 & 174 \end{bmatrix}; \boldsymbol{CB}^{(0)} = \begin{bmatrix} 109 & 109 \\ 109 & 109 \end{bmatrix}$$

Then, one has $SX_L = 77$, $SY_L = 174$ and $SZ_L = 109$ because $SX_L = CR_0$, $SY_L = CG_0$ and $SZ_L = CB_0$, where $L = 2 \times 2 = 4$.

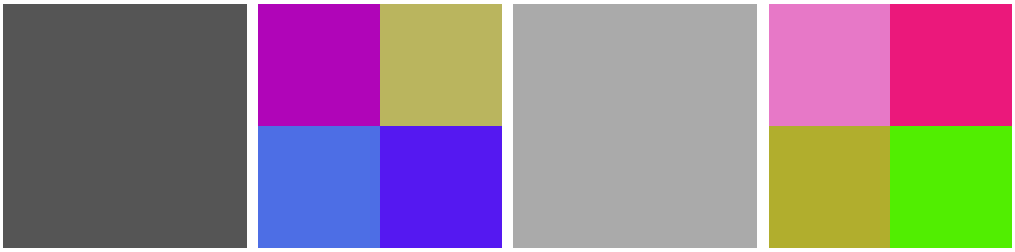
Secondly, based on Step 2 in Section 3.2, choose the two plain images $\boldsymbol{I}^{(85)}$ and $\boldsymbol{I}^{(170)}$, and get the corresponding cipher images, $\boldsymbol{C}^{(85)}$ and $\boldsymbol{C}^{(170)}$, which are shown in Figure 7a–d, respectively. The values of their RGB three channels are:

$$\boldsymbol{IR}^{(85)} = \begin{bmatrix} 85 & 85 \\ 85 & 85 \end{bmatrix}; \boldsymbol{IG}^{(85)} = \begin{bmatrix} 85 & 85 \\ 85 & 85 \end{bmatrix}; \boldsymbol{IB}^{(85)} = \begin{bmatrix} 85 & 85 \\ 85 & 85 \end{bmatrix}$$

$$\boldsymbol{CR}^{(85)} = \begin{bmatrix} 176 & 186 \\ 77 & 85 \end{bmatrix}; \boldsymbol{CG}^{(85)} = \begin{bmatrix} 5 & 181 \\ 110 & 24 \end{bmatrix}; \boldsymbol{CB}^{(85)} = \begin{bmatrix} 184 & 94 \\ 229 & 241 \end{bmatrix}$$

$$\boldsymbol{IR}^{(170)} = \begin{bmatrix} 170 & 170 \\ 170 & 170 \end{bmatrix}; \boldsymbol{IG}^{(170)} = \begin{bmatrix} 170 & 170 \\ 170 & 170 \end{bmatrix}; \boldsymbol{IB}^{(170)} = \begin{bmatrix} 170 & 170 \\ 170 & 170 \end{bmatrix}$$

$$\boldsymbol{CR}^{(170)} = \begin{bmatrix} 231 & 235 \\ 177 & 81 \end{bmatrix}; \boldsymbol{CG}^{(170)} = \begin{bmatrix} 120 & 24 \\ 174 & 238 \end{bmatrix}; \boldsymbol{CB}^{(170)} = \begin{bmatrix} 199 & 123 \\ 45 & 1 \end{bmatrix}$$



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

Figure 7. The two chosen plain images $\boldsymbol{I}^{(85)}$, $\boldsymbol{I}^{(170)}$ and their corresponding cipher images $\boldsymbol{C}^{(85)}$, $\boldsymbol{C}^{(170)}$of size $2 \times 2 \times 3$: **(a)** $\boldsymbol{I}^{(85)}$; **(b)** $\boldsymbol{C}^{(85)}$; **(c)** $\boldsymbol{I}^{(170)}$; **(d)** $\boldsymbol{C}^{(170)}$.

Then, combining Algorithm 1, we determine $\boldsymbol{SX}$ $\boldsymbol{SY}$ $\boldsymbol{SZ}$ as

$$\boldsymbol{SX} = \begin{bmatrix} 84 & 86 & 89 & 77 \end{bmatrix}; \boldsymbol{SY} = \begin{bmatrix} 63 & 31 & 71 & 46 \end{bmatrix}; \boldsymbol{SZ} = \begin{bmatrix} 64 & 36 & 119 & 109 \end{bmatrix}$$

or

$$\boldsymbol{SX} = \begin{bmatrix} 212 & 214 & 217 & 205 \end{bmatrix}; \boldsymbol{SY} = \begin{bmatrix} 191 & 159 & 199 & 174 \end{bmatrix}; \boldsymbol{SZ} = \begin{bmatrix} 192 & 164 & 247 & 237 \end{bmatrix}$$

Thirdly, by Step 3 in Section 3.2, the corresponding permuted image shown in Figure 8c can be restored from the targeted cipher image Figure 8a with $\boldsymbol{SX}$ $\boldsymbol{SY}$ $\boldsymbol{SZ}$. Fourthly, following Step 1 in Section 3.3, construct some special attack images to obtain the permutation matrix $\boldsymbol{PM}$. For images of size $2 \times 2 \times 3$, the process of solving $\boldsymbol{PM}$ is exactly the same as Section 3.3. Then, we determine the $\boldsymbol{PM}$ as Equation (10). Fifth, by Step 2 in Section 3.3, recover $(\boldsymbol{IR}, \boldsymbol{IG}, \boldsymbol{IB})$ from $(\boldsymbol{ER}, \boldsymbol{EG}, \boldsymbol{EB})$ with $\boldsymbol{PM}$. Thus, the

original plain image shown in Figure 8e can be recovered.

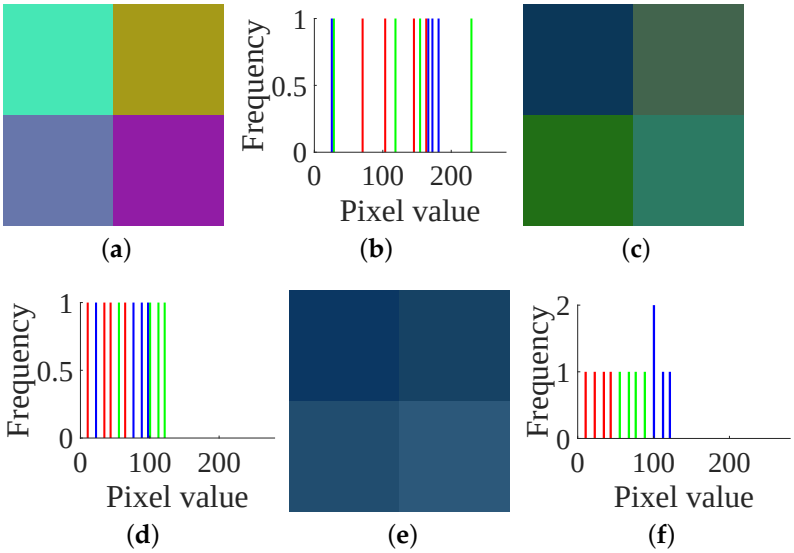


Figure 8. A target cipher image, the permuted image, the original plain image and their histograms of size $2 \times 2 \times 3$: (**a**) a target cipher image; (**b**) histogram of (**a**); (**c**) its permuted image; (**d**) histogram of (**c**); (**e**) its plain image; (**f**) histogram of (**e**).

- *Case 2.* Breaking CIEA-FOHS with "Lenna" of size $256 \times 256 \times 3$:
  Firstly, following Step 1 in Section 3.2, choose the all-zero plain image $\boldsymbol{I}^{(0)}$ shown in Figure 9a and temporarily use the encryption machine of CIEA-FOHS, and then get the corresponding cipher image $\boldsymbol{C}^{(0)}$, as shown in Figure 9b, and the corresponding three channel images and their histograms of $\boldsymbol{C}^{(0)}$ are shown in Figure 9c,d, respectively. Exactly, one has $SX_L = 238$, $SY_L = 168$ and $SZ_L = 91$ owing to $SX_L = CR_0$, $SY_L = CG_0$ and $SZ_L = CB_0$.

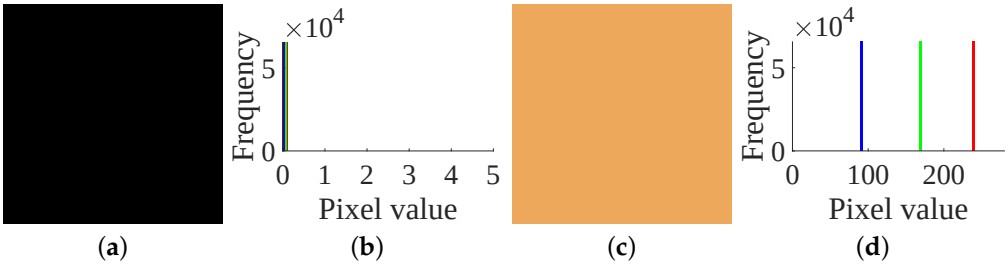


Figure 9. The all-zero chosen plain image $\boldsymbol{I}^{(0)}$ and its corresponding cipher image $\boldsymbol{C}^{(0)}$ of size $256 \times 256 \times 3$: (**a**) $\boldsymbol{I}^{(0)}$; (**b**) histogram of $\boldsymbol{I}^{(0)}$; (**c**) $\boldsymbol{C}^{(0)}$; (**d**) histogram of $\boldsymbol{C}^{(0)}$.

Secondly, based on Step 2 in Section 3.2, choose the two plain images, $\boldsymbol{I}^{(85)}$ and $\boldsymbol{I}^{(170)}$, and get the corresponding cipher images, $\boldsymbol{C}^{(85)}$ and $\boldsymbol{C}^{(170)}$, which are shown in Figure 10a–d, respectively.

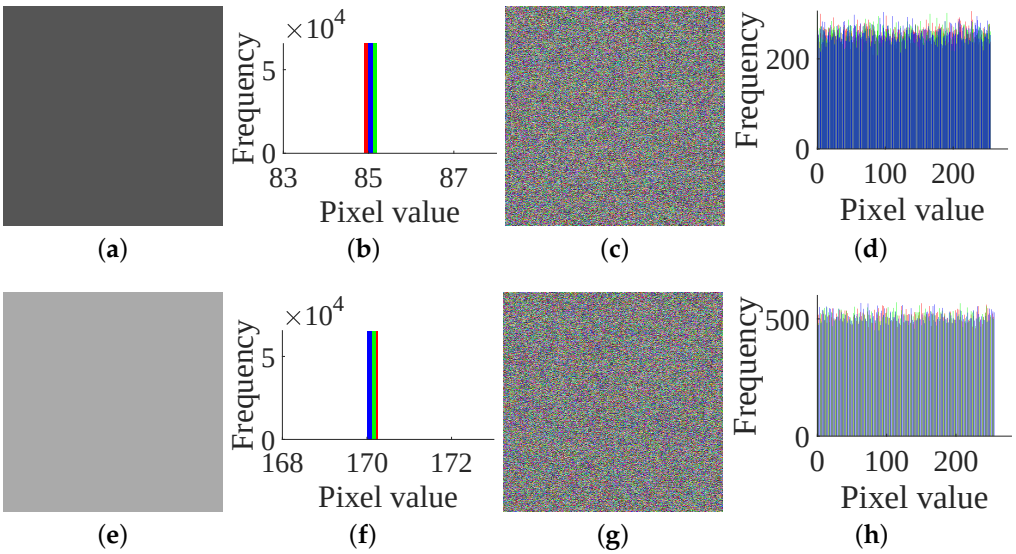


Figure 10. The two chosen plain images $\boldsymbol{I}^{(85)}$, $\boldsymbol{I}^{(170)}$ and their corresponding cipher images $\boldsymbol{C}^{(85)}$, $\boldsymbol{C}^{(170)}$of size $256 \times 256 \times 3$: (**a**) $\boldsymbol{I}^{(85)}$; (**b**) histogram of $\boldsymbol{I}^{(85)}$; (**c**) $\boldsymbol{C}^{(85)}$; (**d**) histogram of $\boldsymbol{C}^{(85)}$; (**e**) $\boldsymbol{I}^{(170)}$; (**f**) histogram of $\boldsymbol{I}^{(170)}$; (**g**) $\boldsymbol{C}^{(170)}$; (**h**) histogram of $\boldsymbol{C}^{(170)}$.

Furthermore, one determines SX_i, SY_i, SZ_i for $i = 1 \sim L-1$ by Algorithm 1. Thirdly, by the method in Section 3.3, choose the three plain images (shown in Figure 11a–f) and get the corresponding cipher images (shown in Figure 11g–l), and then use Algorithm 1 again to obtain their corresponding permuted images (shown in Figure 11m–r). Then, we can get $\boldsymbol{PM}$.

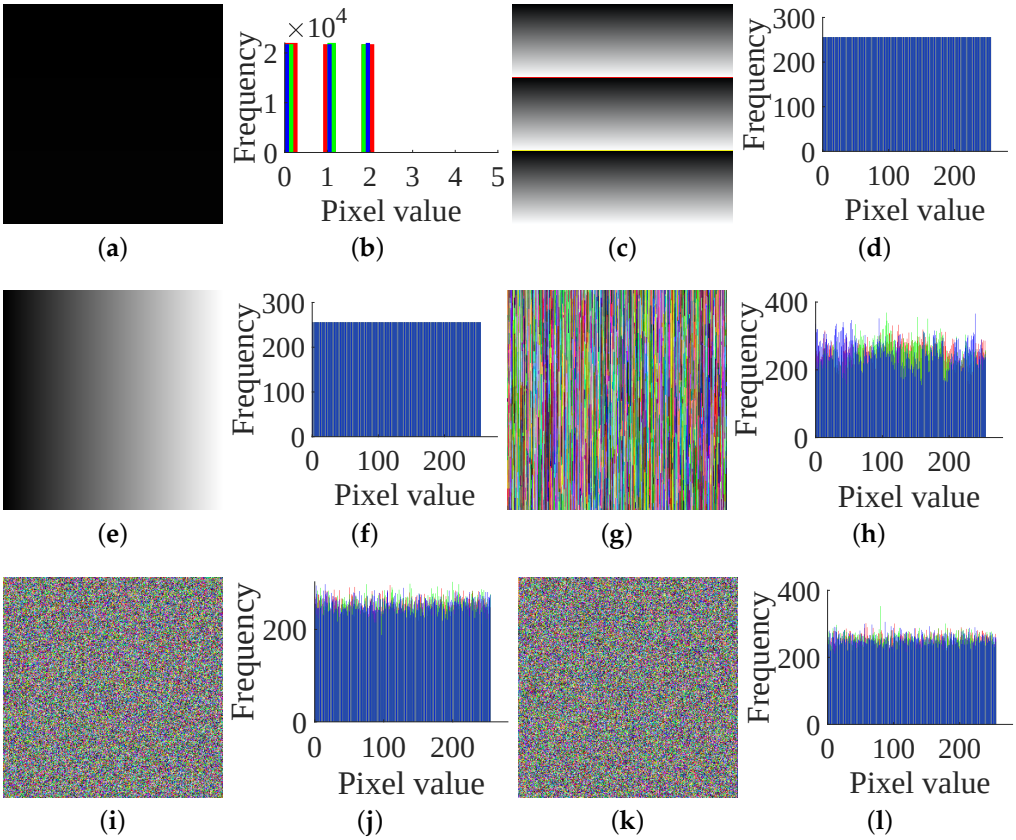


Figure 11. *Cont.*

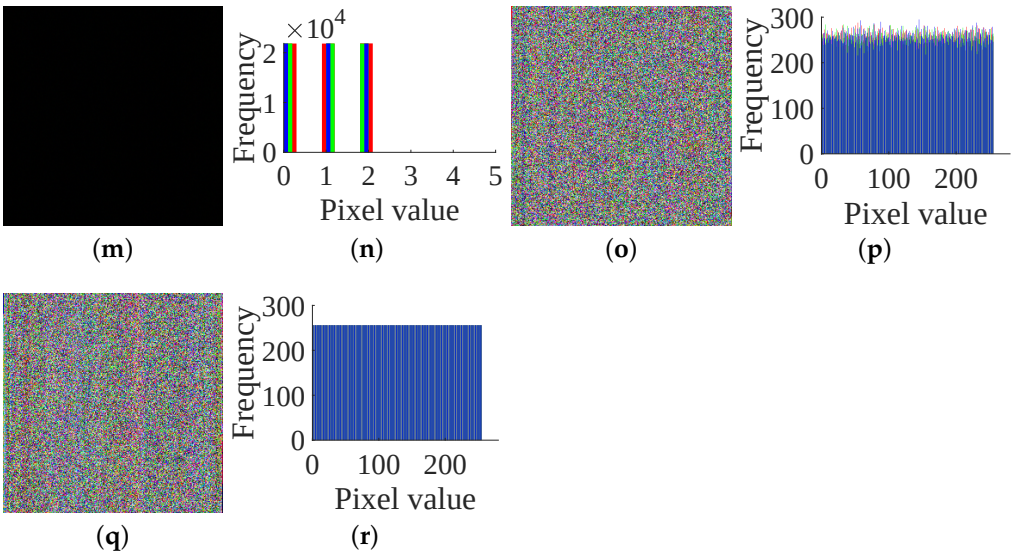


Figure 11. Three chosen plain images, the corresponding cipher and permuted images for attacking permutation: (**a**) $1^{\#}$ plain image; (**b**) The histogram of (**a**); (**c**) $2^{\#}$ plain image; (**d**) The histogram of (**c**); (**e**) $3^{\#}$ plain image; (**f**) The histogram of (**e**); (**g**) $1^{\#}$ cipher image; (**h**) The histogram of (**g**); (**i**) $2^{\#}$ cipher image; (**j**) The histogram of (**i**); (**k**) $3^{\#}$ cipher image; (**l**) The histogram of (**k**); (**m**) $1^{\#}$ permuted image; (**n**) The histogram of (**m**); (**o**) $2^{\#}$ permuted image; (**p**) The histogram of (**o**); (**q**) $3^{\#}$ permuted image; (**r**) The histogram of (**q**).

Finally, we recover the original image from the cipher image of "Lenna" shown in Figure 12a. First, the permuted image shown in Figure 12c is obtained from the cipher image with $(\boldsymbol{SX}, \boldsymbol{SY}, \boldsymbol{SZ})$. Then, the plain image is restored by $\boldsymbol{PM}$, which is shown in Figure 12e.

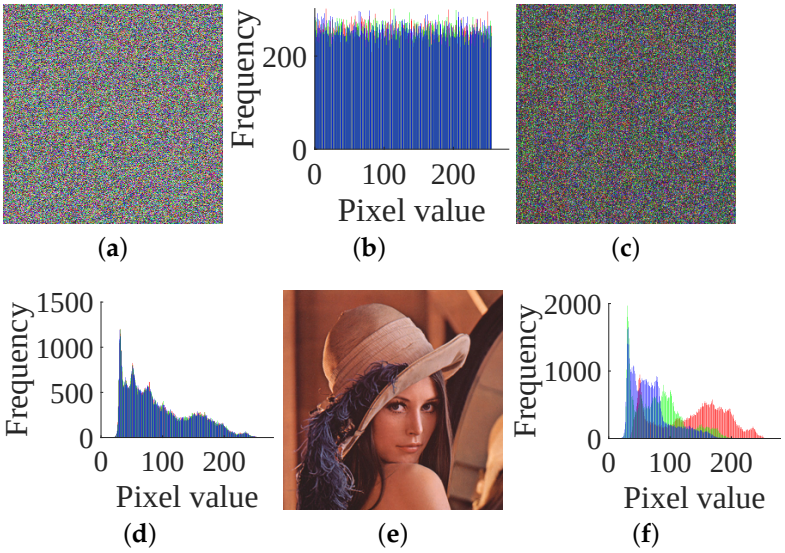


Figure 12. The cipher image, the permuted image, the original plain image of "Lenna" and their histograms of size $256 \times 256 \times 3$: (**a**) the cipher image; (**b**) histogram of (**a**); (**c**) its permuted image; (**d**) histogram of (**c**); (**e**) its plain image; (**f**) histogram of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

Without loss of generality, we do the experiments based on other images with different sizes. The experimental results are shown in Table 3 and Figure 13. They both verify the effectiveness of our attack method. Besides, it can be seen from Table 3 that the proposed attack is efficient. Taking the image "Lenna" of size $256 \times 256 \times 3$ as an example, when the

encryption time is 0.6391 s, the time needed for the corresponding attack is just 129.4039 s. Even if the image size increases, the time required for the attack is still within an acceptable range. Thus, it verifies that our method is computationally feasible.

Moreover, we verified the data complexity required for the attack. As discussed in Section 3.4, the total data complexity required for breaking CIEA-FOHS is $\boldsymbol{O}(3 + \lceil \log_{256}(3HW) \rceil)$. In our experiment with chosen-plaintext attack, the number of attack images required for sizes $2 \times 2 \times 3$ and $100 \times 100 \times 3$ are 4 and 5, respectively. And for sizes $300 \times 200 \times 3$, $256 \times 256 \times 3$ and $512 \times 512 \times 3$, the number of attack images required are all 6. Therefore, the experimental verification is consistent with the theoretical calculation.

Table 3. The time required for breaking CIEA-FOHS by our proposed attack method (unit: second).

Images	Sizes	Encrytion Time	Attacking Diffusion			Attacking Permutation		Totol Attacking Time
			Step 1	*Step* 2	*Step* 3	*Step* 1	*Step* 2	
Figure 5a	$2 \times 2 \times 3$	0.0280	0.1559	0.1811	1.0297	0.0244	2.7151	4.1502
Figure 13b	$100 \times 100 \times 3$	0.1539	0.0920	19.6092	1.1407	0.2764	2.7102	24.0427
Figure 13d	$300 \times 200 \times 3$	0.3280	0.5092	101.7737	0.7872	0.9055	2.4353	106.8545
Figure 12e	$256 \times 256 \times 3$	0.6391	0.6913	120.4768	1.6147	1.9642	3.7725	129.4039
Figure 13f	$512 \times 512 \times 3$	3.5386	2.8134	988.3704	1.9930	4.2884	5.0459	1004.4617

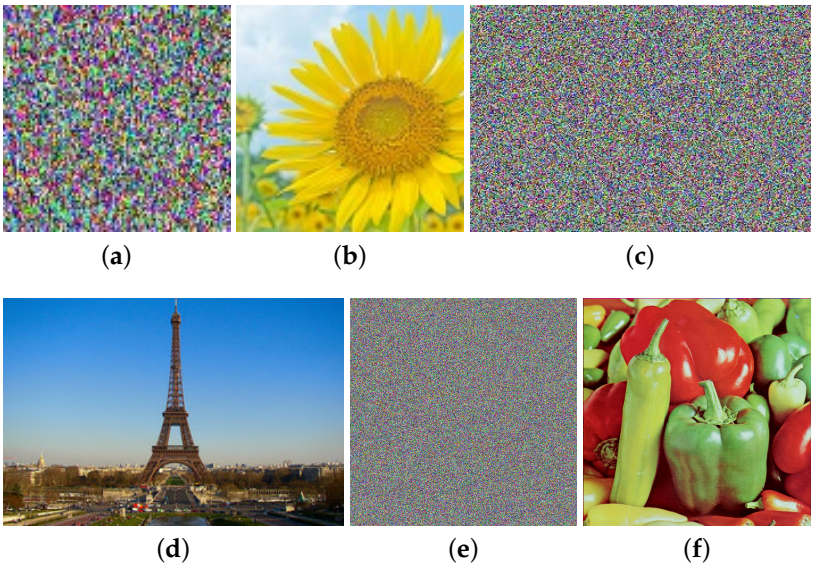


Figure 13. Attacking results with three images of size $100 \times 100 \times 3$, $300 \times 200 \times 3$ and $512 \times 512 \times 3$ respectively: (**a**) cipher image of size $100 \times 100 \times 3$; (**b**) plain image of (**a**); (**c**) cipher image of size $300 \times 200 \times 3$; (**d**) plain image of (**c**); (**e**) cipher image of size $512 \times 512 \times 3$; (**f**) plain image of (**e**).

5. Suggestions for Improvement

On the basis of the above, CIEA-FOHS is insecure against a chosen-plaintext attack method because of its inherent security defects. To enhance the security, some suggestions for improvement are listed below:

- Suggestion 1. Ensuring the substantial security contribution of the fractional-order chaos to the corresponding cipher. The attractor phase diagram of the fractional-order hyperchaotic system is shown in Figure 1, which shows the extremely complex dynamics. Undoubtedly, fractional-order chaos is one of the preferred sources of entropy for encryption. However, due to the negligence of algorithm design, CIEA-FOHS has serious security defects and is attacked.

- Suggestion 2. Security analysis should be implemented from the perspective of cryptography, not limited to numerical statistical verification. As Ref. [45] points out, many encryption algorithms have excellent statistical analysis results, but they are still insecure. In fact, good statistical analysis results are only a necessary and not a sufficient condition for security. Some security flaws are difficult to reflect with numerical statistical results, but they can be clearly revealed by theoretical security analysis. For example, the existence of an equivalent key makes CIEA-FOHS vulnerable to cryptographic attacks. Given the implementation of detailed cryptographic security analysis, these flaws can be avoided, thereby improving security.

6. Conclusions

In this paper, a detailed security analysis of a color image encryption algorithm named CIEA-FOHS using a fractional-order chaos was performed. From the perspective of cryptanalysis, this paper found that CIEA-FOHS can be broken by a chosen-plaintext attack method, owing to its some inherent security defects. Theoretical analysis and experimental simulations show that the attack method is both effective and efficient for attacking CIEA-FOHS. Although the fractional-order chaotic system has complex dynamics, the algorithm defects may cause insecurity. The reported results would help the designers of chaotic cryptography pay more attention to the gap between complex chaotic system and secure cryptosystem.

Author Contributions: Methodology, H.W.; Software, H.W. and J.K.; Validation, H.W., L.H. and C.Z.; Supervision, C.Z.; Project Administration, C.Z. and D.X.; Funding Acquisition, C.Z., H.W. and D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported partly by the National Science Foundation of China (62071088, 61571092), Project for National Key RD Program of China (2018YFB1801302), Project for Innovation Team of Guangdong University (2018KCXTD033), Project for Zhongshan Social Public Welfare Science and Technology (2019B2007), Zhongshan Innovative Research Team Program (180809162197886), Research Project for Talent of UESTC Zhongshan Institute (418YKQN07, 419YKQN23), Natural Science Project for Young Innovative Talents by the Department of Education of Guangdong Province (2019KQNCX191), Characteristic Innovation Project of Department of Guangdong Province (2017GWTSCX010).

Data Availability Statement: No applicable.

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

References

1. Wu, T.; Zhang, C.; Chen, C.; Hou, H.; Wei, H.; Hu, S.; Qiu, K. Security enhancement for OFDM-PON using Brownian motion and chaos in cell. *Opt. Express* **2018**, *26*, 22857–22865. [CrossRef]
2. Wu, T.; Zhang, C.; Chen, Y.; Cui, M.; Huang, H.; Zhang, Z.; Wen, H.; Zhao, X.; Qiu, K. Compressive sensing chaotic encryption algorithms for OFDM-PON data transmission. *Opt. Express* **2021**, *29*, 3669–3684. [CrossRef]
3. Wen, H.; Zhang, C.; Chen, P.; Chen, R.; Xu, J.; Liao, Y.; Liang, Z.; Shen, D.; Zhou, L.; Ke, J. A quantum chaotic image cryptosystem and its application in IoT secure communication. *IEEE Access* **2021**, *1*. [CrossRef]
4. Li, C.; Feng, B.; Li, S.; Kurths, J.; Chen, G. Dynamic analysis of digital chaotic maps via state-mapping networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2019**, *66*. [CrossRef]
5. Li, C.; Tan, K.; Feng, B.; Lu, J. The Graph Structure of the Generalized Discrete Arnold's Cat Map. *IEEE Trans. Comput.* **2021**, *1*. [CrossRef]
6. Li, C.; Zhang, Y.; Xie, E.Y. When an attacker meets a cipher-image in 2018: A Year in Review. *J. Inf. Secur. Appl.* **2019**, *48*. [CrossRef]
7. Akhshani, A.; Akhavan, A.; Mobaraki, A.; Lim, S.C.; Hassan, Z. Pseudo random number generator based on quantum chaotic map. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 101–111. [CrossRef]
8. He, S.; Sun, K.; Wang, H. Complexity analysis and DSP implementation of the fractional-order lorenz hyperchaotic system. *Entropy* **2015**, *17*, 8299–8311. [CrossRef]
9. Shen, C.; Yu, S.; Lü, J.; Chen, G. Designing Hyperchaotic Systems With Any Desired Number of Positive Lyapunov Exponents via A Simple Model. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2014**, *61*, 2380–2389. [CrossRef]
10. Askar, S.S.; Karawia, A.; Al-Khedhairi, A.; Al-Ammar, F.S. An algorithm of image encryption using logistic and two-dimensional chaotic economic maps. *Entropy* **2019**, *21*, 44. [CrossRef]

11. Zhou, Y.; Hua, Z.; Pun, C.; Philip Chen, C.L. Cascade Chaotic System with Applications. *IEEE Trans. Cybern.* **2015**, *45*, 2001–2012. [CrossRef] [PubMed]
12. Wen, H.; Yu, S. Cryptanalysis of an image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. *Eur. Phys. J. Plus* **2019**, *134*, 1–16. [CrossRef]
13. Wen, H.; Yu, S.; Lü, J. Breaking an Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos. *Entropy* **2019**, *21*, 246. [CrossRef]
14. Shafique, A.; Shahid, J. Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. *Eur. Phys. J. Plus* **2018**, *133*, 331. [CrossRef]
15. Song, C.; Qiao, Y. A novel image encryption algorithm based on DNA encoding and spatiotemporal chaos. *Entropy* **2015**, *17*, 6954–6968. [CrossRef]
16. Xie, Y.; Li, C.; Yu, S.; Lü, J. On the cryptanalysis of Fridrich's chaotic image encryption scheme. *Signal Process.* **2017**, *132*, 150–154. [CrossRef]
17. Chen, L.; Ma, B.; Zhao, X.; Wang, S. Differential cryptanalysis of a novel image encryption algorithm based on chaos and Line map. *Nonlinear Dyn.* **2016**, *87*, 1797–1807. [CrossRef]
18. Li, C.; Lin, D.; Lü, J. Cryptanalyzing an Image-Scrambling Encryption Algorithm of Pixel Bits. *IEEE Multimed.* **2017**, *3*, 64–71. [CrossRef]
19. Wang, L.; Sun, K.; Peng, Y.; He, S. Chaos and complexity in a fractional-order higher-dimensional multicavity chaotic map. *Chaos Solitons Fractals* **2020**, *131*, 109488. [CrossRef]
20. Peng, D.; Sun, K.; He, S.; Zhang, L.; Alamodi, A.O.A. Numerical analysis of a simplest fractional-order hyperchaotic system. *Theor. Appl. Mech. Lett.* **2019**, *9*, 220–228. [CrossRef]
21. He, S.; Sun, K.; Wang, H. Dynamics and synchronization of conformable fractional-order hyperchaotic systems using the Homotopy analysis method. *Commun. Nonlinear Sci. Numer. Simul.* **2019**, *73*, 146–164. [CrossRef]
22. Chai, X.; Bi, J.; Gan, Z.; Liu, X.; Zhang, Y.; Chen, Y. Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. *Signal Process.* **2020**, *176*, 107684. [CrossRef]
23. Chai, X.; Wu, H.; Gan, Z.; Han, D.; Zhang, Y.; Chen, Y. An efficient approach for encrypting double color images into a visually meaningful cipher image using 2D compressive sensing. *Inf. Sci.* **2020**, doi:10.1016/j.ins.2020.10.007. [CrossRef]
24. Wang, X.; Chen, S.; Zhang, Y. A chaotic image encryption algorithm based on random dynamic mixing. *Opt. Laser Technol.* **2021**, *138*, 106837. [CrossRef]
25. Hua, Z.; Zhu, Z.; Yi, S.; Zhang, Z.; Huang, H. Cross-plane colour image encryption using a two-dimensional logistic tent modular map. *Inf. Sci.* **2021**, *546*, 1063–1083. [CrossRef]
26. Kamal, F.M.; Elsonbaty, A.; Elsaid, A. A novel fractional nonautonomous chaotic circuit model and its application to image encryption. *Chaos Solitons Fractals* **2021**, *144*, 110686. [CrossRef]
27. Mani, P.; Rajan, R.; Shanmugam, L.; Joo, Y.H. Adaptive control for fractional order induced chaotic fuzzy cellular neural networks and its application to image encryption. *Inf. Sci.* **2019**, *491*, 74–89. [CrossRef]
28. Yang, F.; Mou, J.; Liu, J.; Ma, C.; Yan, H. Characteristic analysis of the fractional-order hyperchaotic complex system and its image encryption application. *Signal Process.* **2020**, *169*, 107373. [CrossRef]
29. Lahdir, M.; Hamiche, H.; Kassim, S.; Tahanout, M.; Kemih, K.; Addouche, S. A novel robust compression-encryption of images based on SPIHT coding and fractional-order discrete-time chaotic system. *Opt. Laser Technol.* **2019**, *109*, 534–546. [CrossRef]
30. Yang, F.; Mou, J.; Ma, C.; Cao, Y. Dynamic analysis of an improper fractional-order laser chaotic system and its image encryption application. *Opt. Lasers Eng.* **2020**, *129*, 106031. [CrossRef]
31. Yu, S.; Zhou, N.; Gong, L.; Nie, Z. Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyper-chaotic system. *Opt. Lasers Eng.* **2020**, *124*, 105816. [CrossRef]
32. Sayed, W.S.; Radwan, A.G. Generalized switched synchronization and dependent image encryption using dynamically rotating fractional-order chaotic systems. *AEU–Int. J. Electron. Commun.* **2020**, *123*, 153268. [CrossRef]
33. Yang, Y.; Guan, B.; Li, J.; Li, D.; Zhou, Y.; Shi, W. Image compression-encryption scheme based on fractional order hyper-chaotic systems combined with 2D compressed sensing and DNA encoding. *Opt. Laser Technol.* **2019**, *119*, 105661. [CrossRef]
34. Fridrich, J. Symmetric Ciphers Based On Two-Dimensional Chaotic Maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284. [CrossRef]
35. Zhao, J.; Wang, S.; Chang, Y.; Li, X. A novel image encryption scheme based on an improper fractional-order chaotic system. *Nonlinear Dyn.* **2015**, *80*, 1721–1729. [CrossRef]
36. Norouzi, B.; Mirzakuchaki, S. Breaking a novel image encryption scheme based on an improper fractional order chaotic system. *Multimed. Tools Appl.* **2017**, *76*, 1817–1826. [CrossRef]
37. Ye, G. Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognit. Lett.* **2010**, *31*, 347–354. [CrossRef]
38. Zhou, G.; Zhang, D.; Liu, Y.; Yuan, Y.; Liu, Q. A novel image encryption algorithm based on chaos and Line map. *Neurocomputing* **2015**, *169*, 150–157. [CrossRef]
39. Huang, X.; Zhao, Z.; Wang, Z.; Li, Y. Chaos and hyperchaos in fractional-order cellular neural networks. *Neurocomputing* **2012**, *94*, 13–21. [CrossRef]
40. Wang, Z.; Huang, X.; Li, Y.; Song, X. Image encryption based on a delayed fractional-order chaotic logistic system. *Chin. Phys. B* **2013**, *22*, 010504. [CrossRef]

41. Zhang, L.; Sun, K.; Liu, W.; He, S. A novel color image encryption scheme using fractional-order hyperchaotic system and DNA sequence operations. *Chin. Phys. B* **2017**, *26*, 100504. [CrossRef]
42. Huang, X.; Sun, T.; Li, Y.; Liang, J. A Color Image Encryption Algorithm Based on a Fractional-Order Hyperchaotic System. *Entropy* **2015**, *17*, 28–38. [CrossRef]
43. Li, C.; Liu, Y.; Zhang, L.Y.; Chen, M.Z.Q. Breaking a chaotic image encryption algorithm based on modulo addition and XOR operation. *Int. J. Bifurc. Chaos* **2013**, *23*, 1350075. [CrossRef]
44. Zhang, L.Y.; Liu, Y.; Pareschi, F.; Zhang, Y.; Wong, K.; Rovatti, R.; Setti, G. On the Security of a Class of Diffusion Mechanisms for Image Encryption. *IEEE Trans. Cybern.* **2018**, *48*, 1163–1175. [CrossRef] [PubMed]
45. Preishuber, M.; Hütter, S.K.T.; Uhl, A. Depreciating Motivation and Empirical Security Analysis of Chaos-Based Image and Video Encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2137–2150. [CrossRef]