Hindawi
Complexity
Volume 2019, Article ID 7485621, 16 pages
https://doi.org/10.1155/2019/7485621

Research Article

Image Encryption Based on Dynamic Filtering and Bit Cuboid Operations

Xinsheng Li,[1] Zhilong Xie,[2] Jiang Wu,[2] and Taiyong Li [2,3]

[1] *School of Computer Sciences, Sichuan University, Chengdu 610064, China*
[2] *School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 611130, China*
[3] *Sichuan Province Key Laboratory of Financial Intelligence and Financial Engineering, Southwestern University of Finance and Economics, Chengdu 611130, China*

Correspondence should be addressed to Taiyong Li; litaiyong@gmail.com

Received 12 November 2018; Revised 4 January 2019; Accepted 9 January 2019; Published 3 February 2019

Guest Editor: Amir Anees

As one of the most widely used media types, images play an important role in the era of the Internet. And hence how to enhance the security of images has become a hot topic in the field of information security. However, due to some intrinsic characteristics of images, image security is still a challenging task. For the purpose of coping with this issue, in this paper, we propose a novel algorithm that combines a hyperchaotic system, dynamic filtering, and bit cuboid operations, namely, DFBC, for image encryption. Specifically, the proposed DFBC consists of four steps: firstly, a 7D Lorenz hyperchaotic system is utilized to generate a pseudorandom sequence; secondly, variable 1D filters are derived from the pseudorandom sequence, and dynamic filtering is conducted on each pixel of an image; thirdly, a diffusion scheme is performed and then the image is transformed to a bit cuboid; and, finally, various types of permutation (rearranging, symmetry, rotation, zigzag, and global bit permutation) are performed on the bit cuboid. The experiments on several testing images demonstrate that the DFBC achieves state-of-the-art results in terms of several evaluation criteria, showing that the DFBC is promising for image encryption.

1. Introduction

Image security is one of the most important types of information security. In recent years, image encryption has become a hot topic in the field of image security. However, since images usually show some intrinsic properties such as bulky data capacity, high redundancy, and strong correlation, traditional encryption algorithms such as data encryption standard (DES), advanced encryption standard (AES), and international data encryption algorithm (IDEA) for common data are not able to be applied to images directly to achieve good results [1, 2]. To cope with this issue, various researchers have been devoted to proposing image encryption algorithms to enhance image security in recent years.

Among the image encryption algorithms, chaos-based ones have become more and more popular in recent years because chaotic systems have some special properties for image encryption, such as pseudorandomness, ergodicity, unpredictability, and extreme sensitivity to system parameters and initial values [3–7]. Generally speaking, there are two ways to perform image encryption, i.e., permutation and diffusion. The former means changing the position of image data, while the latter means changing the values of image data. Many practical image encryption algorithms use these two approaches together. When chaotic systems are applied to image encryption, the common ways are to perform permutation and diffusion using the index and the values of chaotic sequences generated from chaotic systems, respectively. Chen et al. proposed a novel real-time secure symmetric image encryption scheme that generalized a 2D chaotic map to 3D and then used the 3D cat map and another chaotic map to conduct permutation and diffusion, respectively [3]. Zhang proposed a novel fast image cryptosystem that used the piecewise linear chaotic map and cubic S-box to generate the sequence for image encryption [6]. Typically, low-dimensional chaotic systems have simple forms and are

easy to implement, and hence some scholars applied them to image encryption [8, 9].

However, low-dimensional chaotic maps usually have only a few variables and parameters, along with simple structures and chaotic orbits, making it easy to estimate the orbits and the initial parameters, and hence the security of image is reduced [2, 10]. Therefore, higher-dimensional chaotic systems have been widely applied to image encryption. Some researchers apply 3D chaotic maps to image encryption [11, 12]. In a dynamical system, the Lyapunov exponent (LE) is a quantity that characterizes the rate of separation of infinitesimally close trajectories and estimates the chaos of the system [13], and if a chaotic system has two or more positive LEs, the system is defined to be hyperchaotic [14]. Recent research has demonstrated that image encryption algorithms associated with hyperchaotic systems show greater security [15–23]. Among the approaches, 4-6D hyperchaotic systems are widely used to enhance image encryption. Zhu and Sun proposed an image encryption algorithm with two-round encryption operations based on a 4D hyperchaotic system [16]. Xue et al. applied a 5D hyperchaotic system with 3 positive LEs for region of interest encryption for color images [21]. Wu et al. presented a new lossless encryption algorithm that used the 2D discrete wavelet transform (DWT) and a 6D hyperchaotic system in both frequency and spatial domains for color images, and the experimental results indicated that the proposed algorithm was effective and efficient [23]. Generally speaking, the more the number of positive LEs or the higher the dimension of a hyperchaotic system is, the more secured the hyperchaotic system-based image encryption algorithms are.

When an image encryption algorithm is performed on an image, it usually processes block-level data (a block of pixels) [24–27], pixel-level data [28, 29], deoxyribonucleic acid-(DNA-) level data (2 bits) [30–33], bit-level data [34–36], or bit plane-level data [37, 38]. For fixed processing power, the lower the processing level is, the more pixels the algorithm can handle in one time. For example, if the algorithm can process 8 bits in one time for 256-level gray images, it means that it can process 1 pixel, 4 nucleic acid bases, or 8 bits, which involves 1/4/8 pixels at the most, respectively. Therefore, encryption algorithms associated with low-level data usually show better performance in image encryption.

Image filtering, also known as convolution, is one of the most important operations in image processing, and its typical applications include denoising, smoothing, and edge detection. Most recently, Hua and Zhou have applied this technique to image encryption for the first time, and they proposed an approach called block-based scrambling and image filtering (IC-BSIF) for image encryption. The experimental results showed that it could achieve better performance than some state-of-the-art encryption schemes [26]. In this scheme, the authors use a fixed template of filter for all pixels, losing the diversity of the template and hence not maximizing the performance of the scheme. In the field of chaos, Yang et al. have found a new 7D autonomous hyperchaotic system with 5 positive LEs recently, which has very simple algebraic structure but can show complex dynamical behaviors [39]. Therefore, it has good potential for image encryption.

Motivated by the above analysis, in this paper, we aim to propose a novel image encryption scheme that combines a 7D hyperchaotic system with 5 positive LEs, dynamic filtering, and bit cuboid operation, namely, DFBC, for image encryption. The main steps of the DFBC are as follows. (1) A 7D hyperchaotic system is used to generate a pseudorandom sequence for subsequent encryption operations. (2) 1D dynamic filtering is conducted on each pixel with random filters derived from the pseudorandom sequence. (3) The 2D image of pixel plane is transformed to a 3D bit cuboid. (4) Various types of permutation, such as rearranging, symmetry, rotation, zigzag, and global bit permutation, are performed on the bit cuboid.

The rest of this paper is structured as follows. A brief description of the 7D hyperchaotic system and filtering is given in Section 2. In Section 3, the novel image encryption algorithm, DFBC, is proposed in detail. Experimental results are reported in Section 4. Finally, we conclude the paper in Section 5.

2. Preliminaries

2.1. Hyperchaotic Systems. As one of the most popular chaotic systems, the Lorenz chaotic system and its extensions are very popular in image encryption [40–44]. Most recently, Yang et al. have found a new 7D autonomous hyperchaotic system with 5 positive LEs by adding feedback controllers to the Lorenz system [39], formulated as

$$\begin{aligned}
\dot{x}_1 &= a\left(x_2 - x_1\right) + x_4 + bx_6,\\
\dot{x}_2 &= cx_1 - x_2 - x_1x_3 + x_5,\\
\dot{x}_3 &= -dx_3 + x_1x_2,\\
\dot{x}_4 &= ex_4 - x_1x_3,\\
\dot{x}_5 &= -fx_2 + x_6,\\
\dot{x}_6 &= gx_1 + hx_2,\\
\dot{x}_7 &= ix_7 + jx_4,
\end{aligned} \tag{1}$$

where $adefi \neq 0$, a, c, d are the constant parameters, b, e, f, g, h, and i are control parameters, and j is the coupling parameter. When the parameters $(a, b, c, d, e, f, g, h, i, j) = (10, 1, 28, 8/3, 2, 9.9, 1, 2, 1, 1)$ and initial values $(x_1^0, x_2^0, x_3^0, x_4^0, x_5^0, x_6^0, x_7^0) = (0, 0.1, 0.2, 0.3, 0.5, 0.6, 0.7)$, the attractors of the 7D hyperchaotic system are shown in Figure 1 (the first 1000 values of each component are removed to eliminate the adverse effects). Correspondingly, keeping the parameters except for f unchanged, the LE spectrum of the 7D hyperchaotic system with respect to parameter $f \in (0, 80]$ is shown in Figure 2. It can be seen that, when $f \in [0, 67.4]$ (approximately), the system has hyperchaotic behaviors because it has two or more positive LEs. In particular, when $f \in [6.5, 16.4]$ (approximately), the 7D hyperchaotic system has five positive LEs.

The advantages of this hyperchaotic system are two aspects: (1) the algebraic structure is very simple, so it is easy to implement and (2) the system exhibits complex dynamical

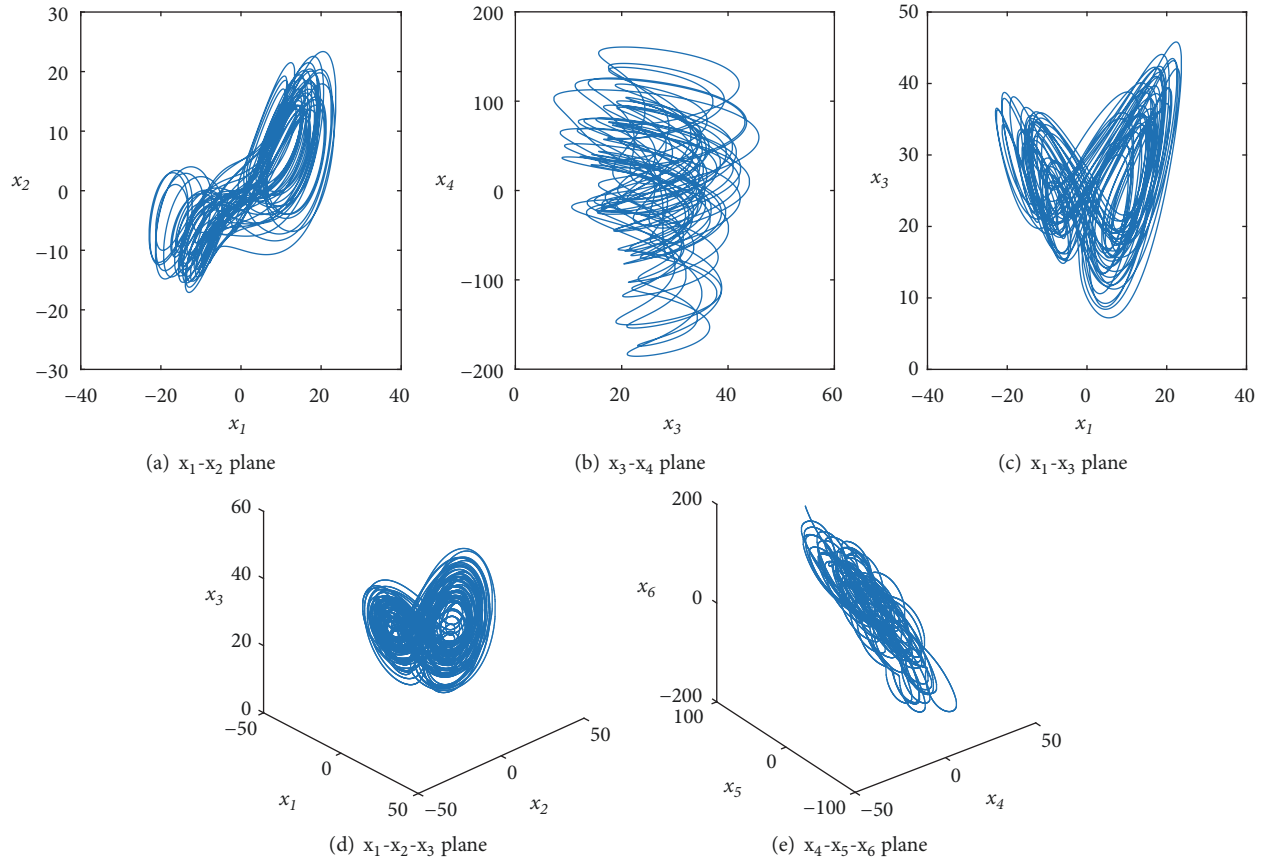


Figure 1: Hyperchaotic attractor of the 7D Lorenz hyperchaotic system (1).

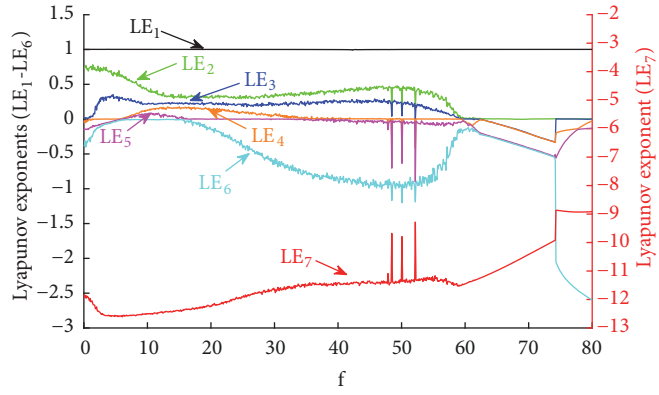


Figure 2: Lyapunov exponent spectrum of the 7D Lorenz hyperchaotic system (1).

behaviors, and hence it is suitable for image encryption. Therefore, in this paper, we use this hyperchaotic system for image encryption.

2.2. Filtering Operation. It is well-known that image filtering by convolution is widely used in many digital image processing fields, such as adding soft blur, sharpening details, accentuating edges, or denoising. Some of filtering is linear, which involves weighted combinations of pixels in small neighborhoods. The simplest filter to implement is the moving average or box filter, which simply averages the pixel values in an $X \times X$ window. Usually, the gradient, Laplacian, and difference of Gaussians (DoG) filters work very well for general purpose visual enhancement or edge detection. This is equivalent to convolution of the image with a mask of all ones.

Traditional image filtering cannot be reversed; that is to say, it usually cannot recover the original image with the filtered image and the filters. To overcome this shortcoming, very recently, Hua and Zhou have designed a special filter with which the original image can be recovered from the filtered image. And they have proposed a novel approach called IC-BSIF for image encryption, where image filtering

is applied to image encryption for the first time [26]. The experimental results demonstrated the effectiveness of the proposed IC-BSIF.

2.3. Bit-Level Operation. Permutation conducted on pixel-level data or higher level data (e.g., blocks of pixels) cannot change the statistical characteristics of an image [45], while permutation on bit-level data can change such characteristics. Furthermore, for a fixed process width (bits processed in one run), the bit-level image encryption can involve more pixels than higher level data (e.g., DNA-level, pixel-level, or blocks of pixels), for both permutation and diffusion. Therefore, to enhance the security of image encryption and to speed up the algorithm, in this paper, we will perform bit-level encryption. Specifically, we first transform an image into a cuboid that is very close to a cube, and then various bit plane operations are conducted on the cuboid.

3. The Proposed Image Encryption Approach

3.1. Hyperchaotic Sequence Generation. Since the 7D hyperchaotic system described in Section 2.1 has good properties for image encryption, we use it to generate the hyperchaotic sequence applied in the proposed image encryption approach. The generating process consists of three steps.

Step 1. To eliminate the adverse effects, the 7D hyperchaotic system is firstly iterated N_0 times and then the sequence generated is removed.

Step 2. The 7D hyperchaotic system continues to iterate until the generated sequences are long enough to image encryption. For the j–th iteration, seven state values denoted by $s^j = \{x_1^j, x_2^j, \cdots, x_7^j\}$ are obtained.

Step 3. After the whole iteration, the hyperchaotic sequences K can be obtained by contacting all the s^j ($j = 1, 2, \cdots, N$) as

$$\begin{aligned} K &= \left\{s^1, s^2, \cdots, s^N\right\} \\ &= \left\{x_1^1, x_2^1, \cdots, x_7^1, \cdots, x_1^N, x_2^N, \cdots, x_7^N\right\} \\ &= \left\{k_1, k_2, k_3, \cdots, k_{7N-2}, k_{7N-1}, k_{7N}\right\}. \end{aligned} \tag{2}$$

The purposes of the generated sequence K for encryption are two aspects: (1) sorting subsequence of K to get the index of original data for permutation; (2) using subsequence of K to change the values of images for diffusion. In our approach, we map the i–th point in K to the integral range of [0,255] by (3) and to the real range of [0, 1) by (4) for the first purpose and the second purpose, respectively.

$$k_i = \mathrm{mod}\left(\left\lfloor \mathrm{mod}\left(\left(|k_i| - \lfloor |k_i| \rfloor\right) \times 10^{15}\right), 10^8\right)\right\rfloor, 256\Big), \tag{3}$$

$$k_i = \left|k_i \times 10^8 - \left\lfloor k_i \times 10^8 \right\rfloor\right|, \tag{4}$$

where mod is the modulo operation, $|\cdot|$ is the absolute value operation, and $\lfloor\cdot\rfloor$ denotes flooring operation [2].

3.2. Dynamic Filtering

3.2.1. Image Filtering. Image filtering is the convolution operation between an image and a template W, also known as a mask. This template is usually a 2D matrix whose size is 3×3, 5×5, 7×7, etc. Suppose the size of a 2D template is $(2M + 1) \times (2M + 1)$, the convolution operation of image filtering can be defined as

$$R_{x,y} = \sum_{i=-M}^{M} \sum_{j=-M}^{M} W_{i+M+1, j+M+1} I_{x+i, y+j}. \tag{5}$$

If the template is a one-row or one-column matrix, it is called 1D convolution, which can be defined as the following:

$$R_{x,y} = \sum_{i=-M}^{M} W_{i+M+1} I_{x, y+i}, \tag{6}$$

$$R_{x,y} = \sum_{i=-M}^{M} W_{i+M+1} I_{x+i, y}, \tag{7}$$

where the sizes of W in (6) and (7) are $1 \times (2M + 1)$ and $(2M + 1) \times 1$, respectively.

3.2.2. 1D Dynamic Filtering. Hua and Zhou used 2D filtering to do diffusion in image encryption. They also introduced Proposition 1 and its proof to identify that image filtering operation satisfying some conditions can be reversible [26].

Proposition 1. *For any given mask W of size $(2M + 1) \times (2M + 1)$, image I of size $X \times Y$, and P' grayscale level F, the operation*

$$R_{x,y} = \left(\sum_{i=-M}^{M} \sum_{j=-M}^{M} W_{i+M+1, j+M+1} I_{x+i, y+j}\right) \bmod F, \tag{8}$$

can be reversible and its inverse operation is

$$I_{x,y} = \left(R_{x,y} - \sum_{i=-M, i\neq 0}^{M} \sum_{j=-M, j\neq 0}^{M} W_{i+M+1, j+M+1} R_{x+i, x+j}\right) \cdot \bmod F, \tag{9}$$

if $P \in \mathbb{N}, W \in \mathbb{N}$ and $W_{M+1,M+1} = 1$.

The key difference to the ordinary image filter is to set the mask center to 1; i.e., $W_{M+1,M+1} = 1$.

Here we can prove that if the center of 1D image filter is set to 1, Proposition 1 still holds. And it is rewritten as Proposition 2.

Proposition 2. *For any given 1D mask W of size $(2M + 1) \times 1$, image I of size $X \times Y$, and P's grayscale level F, the operation*

$$R_{x,y} = \left(\sum_{i=-M}^{M} W_{i+M+1} I_{x+i, y}\right) \bmod F, \tag{10}$$

is reversible and its inverse operation is

$$I_{x,y} = \left(R_{x,y} - \sum_{i=-M, i\neq 0}^{M} W_{i+M+1} R_{x+i, y}\right) \bmod F, \tag{11}$$

if $P \in \mathbb{N}, W \in \mathbb{N}$ and $W_{M+1} = 1$.

For the mask with direction along with coordinate y, or, in other words, 1D mask W of size $1 \times (2M + 1)$, 1D image filtering has the same property.

Proof. Since $W_{M+1} = 1$, (10) can be rewritten as

$$\begin{aligned} R_{x,y} &= \left(\left(\sum_{i=-M, i\neq 0}^{M} W_{i+M+1} I_{x+i,y} \right) + W_{M+1} I_{x,y} \right) \bmod F \\ &= \left(\left(\sum_{i=-M, i\neq 0}^{M} W_{i+M+1} I_{x+i,y} \right) + 1 \times I_{x,y} \right) \bmod F \\ &= \left(\sum_{i=-M, i\neq 0}^{M} W_{i+M+1} I_{x+i,y} \right) + I_{x,y} - kF, \end{aligned} \tag{12}$$

where k is an integer. Reshape (12); then it becomes

$$I_{x,y} = R_{x,y} + kF - \sum_{i=-M, i\neq 0}^{M} W_{i+M+1} I_{x+i,y}. \tag{13}$$

Now (13) can be rewritten as follows because $R \in \mathbb{N}$ and $0 \leq R_{x,y} < F$.

$$\begin{aligned} I_{x,y} &= \left(R_{x,y} + kF - \sum_{i=-M, i\neq 0}^{M} W_{i+M+1} I_{x+i,y} \right) \bmod F \\ &= \left(R_{x,y} - \sum_{i=-M, i\neq 0}^{M} W_{i+M+1} I_{x+i,y} \right) \bmod F. \end{aligned} \tag{14}$$

□

By now, the proof of Proposition 2 has been completed. It has the similar formation with (9). As we can see that, for Proposition 2, even if every pixel $I_{x,y}$ is filtered by different template $W^{x,y}_{i+M+1}$, Proposition 2 still holds and then is detailed as Proposition 3 as follows.

Proposition 3. *For any given mask $W^{x,y}$ of size $(2M + 1) \times 1$ at pixel $I_{x,y}$ of P's grayscale level F, the operation*

$$R_{x,y} = \left(\sum_{i=-M}^{M} W^{x,y}_{i+M+1} I_{x+i,y} \right) \bmod F, \tag{15}$$

is reversible where image size is size $X \times Y$ and its inverse operation is

$$I_{x,y} = \left(R_{x,y} - \sum_{i=-M, i\neq 0}^{M} W^{x,y}_{i+M+1} R_{x+i,y} \right) \bmod F, \tag{16}$$

if $P \in \mathbb{N}, W^{x,y} \in \mathbb{N}$ and $W^{x,y}_{M+1} = 1$.

The logic of setting our masks $W^{x,y}_{i+M+1}$ is described in Figure 3.

The colored $W^{x,y}_{M+1}$ is the center of the mask of filtering which is 1 exactly. It pairs with the pixel $I_{x,y}$ which needs to be filtered. The other weights in the mask after the colored $W^{x,y}_{M+1}$ are set to 0. Then this $2M + 1$ mask is simplified to $M + 1$ and the weight with respect to the pixel $I_{x,y}$ is the last weight in the mask. In the proposed approach, the mask $W^{x,y}$ is said to be a dynamic one because, for each pixel $I_{x,y}$, we use different values from the hyperchaotic sequence to fill the mask as the weights.

When filtering is conducted on the edge of an image, the original image has to be expanded [26]. Since the filter is 1D, we only need to expand the image on horizontal or vertical direction. When horizontal filter is working, the most right M columns are copied to the left side of the image. Similarly, when vertical filter is working, the most bottom M rows are copied to the top side of the image. Of course, the number of rows or columns which need to be copied in filtering process has to change accordingly at the row or column between 1 and M. This process is depicted by an example in Figure 4. The mask size in the example is set as 3×1 and 1×3.

Our dynamic diffusion mask of filtering comes from the hyperchaotic sequence. And the indicator of direction of filtering mask is determined by one number in the hyperchaotic sequence too. If the number is even, a horizontal mask is applied to the image. Otherwise, an odd number means a vertical mask.

After the dynamic masks are set, the convolution begins as the normal convolution does except that the mask has to be renewed for each pixel.

3.3. Global Pixel Diffusion. Although the dynamic filtering is capable of diffusing images, to further enhance the performance of the diffusion of our approach, we conduct a simple two-step diffusion for image with pixels after dynamic filtering, following the diffusion scheme in [2]. Given an image of size $L = X \times Y$, where X and Y are the height and width, respectively, the scheme firstly transforms the image into a 1D pixel sequence $P = \{p^i\}$, $i = 1, 2, \cdots, L$. Given an initial key k^0 and a chaotic key sequence $K = \{k^i \in [0, 255]\}$, $i = 1, 2, \cdots, L$, the diffusion can be conducted in two steps, as formulated in the following equations, respectively:

$$\begin{aligned} q^1 &= p^1 \otimes \bmod \left(k^0 + k^1, 256 \right), \\ q^i &= p^i \otimes \bmod \left(q^{i-1} + k^i, 256 \right), \end{aligned} \tag{17}$$

$$\begin{aligned} q^1 &= q^1 \otimes \bmod \left(\left| q^L - k^1 \right|, 256 \right), \\ q^i &= q^i \otimes \bmod \left(\left| q^{i-1} - k^i \right|, 256 \right), \end{aligned} \tag{18}$$

where $\otimes$ is the XOR operation and q^i is i−th pixel in the diffused image.

Besides the dynamic filtering and global pixel diffusion, we also conduct the XOR operation on the pixels of image with a random mask to change the value of the pixels directly.

3.4. Bit Cuboid Operations

3.4.1. Pixel Plane to Bit Cuboid. Given a plain grayscale image with X rows (height) and Y columns (width), it is a typical

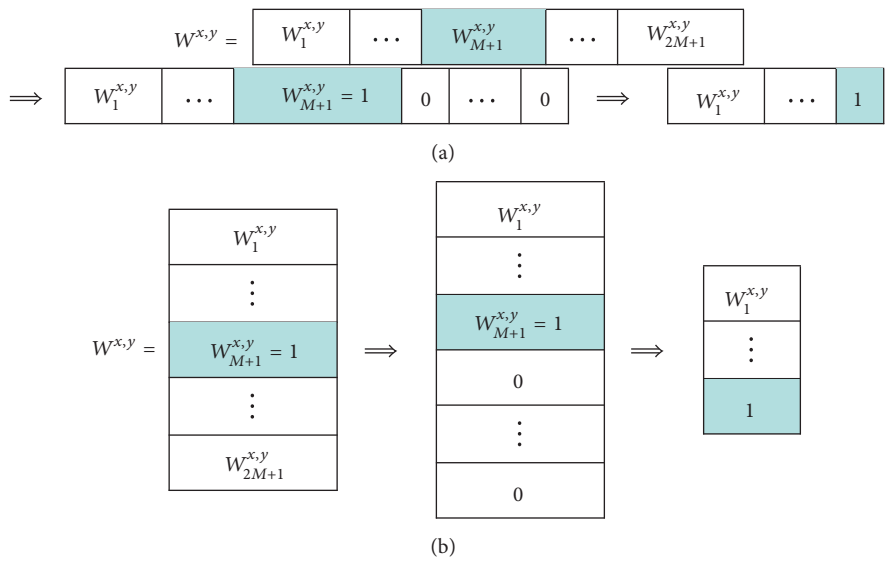


FIGURE 3: The horizontal and vertical masks are deduced from the original ones with size $2M + 1$ to the simplified ones with size $M + 1$. (a) Horizontal 1D mask. (b) Vertical 1D mask.

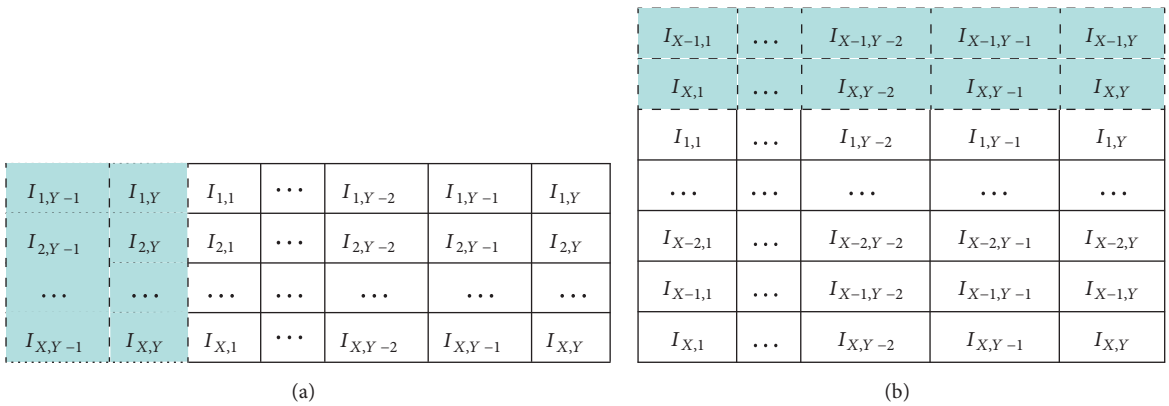


FIGURE 4: Demonstration of handling edge of an image. (a) Horizontally expanding when the first column is processing. (b) Vertically expanding when the first row is processing. The mask size is 3.

pixel plane with $X \times Y$ pixels, each of which is one 256-level value. The total bits of the image are $X \times Y \times 8$, so it is naturally a cuboid. To perform bit-level operations better, the cuboid should be transformed into a new one whose width, height, and depth can be as uniform as possible. To achieve this, we transform the original cuboid with size of $X \times Y \times 8$ into a new one with size of $X' \times Y' \times d$, as

$$X' = Y' = 2^{\lfloor \log_2(\sqrt[3]{X \times Y \times 8}) \rfloor},$$
$$d = \frac{X \times Y \times 8}{X' \times Y'}. \tag{19}$$

Equation (19) can ensure that both X' and Y' are integers. For the simplicity, in this paper, we do not consider the cases when d is not integral. With this equation, the original pixel plane can be transformed into a cuboid which has as uniform a side length as possible. For example, grayscale images having sizes of 512×512 and 256×256 can be transformed into two cuboids with sizes of $128 \times 128 \times 128$ and $64 \times 64 \times 128$, respectively.

3.4.2. Operations on Bit-Planes

(1) Bit Plane Rearranging. Bit plane rearranging can be conducted at different directions for a bit cuboid. Since the operations at each direction are almost the same, here we use the operation at the direction of depth as an illustration for simplicity. The rearranging steps are as follows.

Step 1. Given a bit cuboid of $h \times w \times d$ and a random sequence $S = \{s_i\}$, $i = 1, 2, \cdots, d$, sort S in ascending order to get the index sequence p_i, $i = 1, 2, \cdots, d$.

Step 2. According to the index sequence, rearrange the original i-th bit plane at the direction of depth as the p_i-th bit plane in the rearranged bit cuboid.

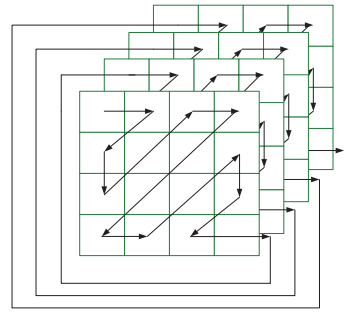


Figure 5: An illustration of zigzag.

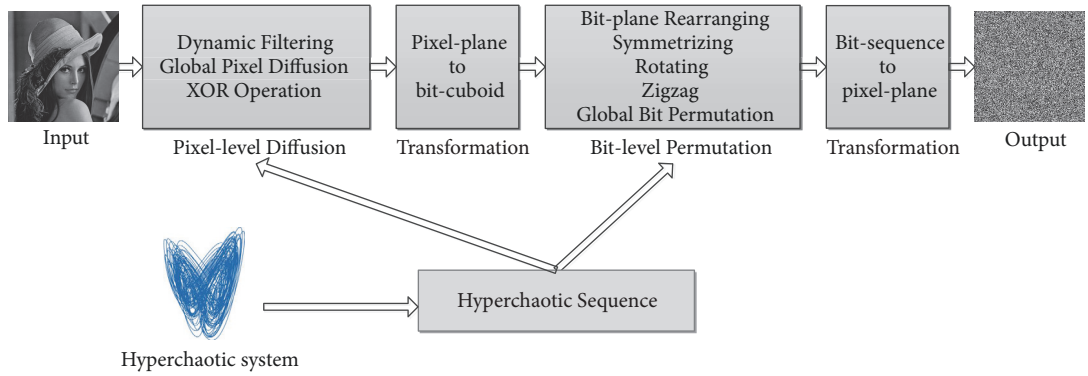


Figure 6: The framework of the DFBC.

(2) Symmetry. The operations of symmetry include two types: horizontal symmetry and vertical symmetry. For a given bit plane $B = \{b(i, j)\}$, $i = 1, 2, \cdots, I$, $j = 1, 2, \cdots, J$, the horizontal symmetry is formulated as $b(i, j) \longleftrightarrow b(I-i+1, j)$, $i = 1, 2, \cdots, \lfloor I/2 \rfloor$, $j = 1, 2, \cdots, J$, and the vertical symmetry is defined as $b(i, j) \longleftrightarrow b(i, J - j + 1)$, $i = 1, 2, \cdots, I$, $j = 1, 2, \cdots, \lfloor J/2 \rfloor$.

(3) Rotation. Since a bit plane may be a rectangle (not always a square), the angle of this rotation can only be 180 degrees. Therefore, For a given bit plane $B = \{b(i, j)\}$, $i = 1, 2, \cdots, I$, $j = 1, 2, \cdots, J$, the operation of rotation is defined as $b(i, j) \longleftrightarrow b(I - i + 1, J - j + 1)$, $i = 1, 2, \cdots, \lfloor I/2 \rfloor$, $j = 1, 2, \cdots, J$.

(4) Zigzag. Zigzag can disturb the high correlation for images to enhance the security of encrypted images [46]. In this paper, we use zigzag to permutate the bit-planes in the bit cuboid one by one, as illustrated in Figure 5. In each bit plane, the zigzag confusion path starts from the upper left bit and ends at the bottom right bit. When scanning a plane is completed, the scan starts from the upper left bit of the next plane, and so on. When all the bit-planes are scanned, a bit string is obtained. Zigzag can not only permute the bits, but also diffuse corresponding pixel-level data.

(5) Global Bit Permutation. After Zigzag, the encryption scheme will obtain a bit sequence. The global bit permutation is to permute all bits in the bit sequence with a hyperchaotic sequence [2, 30]. The steps are as follows.

Step 1. Given a bit sequence of length L and a hyperchaotic sequence $S = \{s_i\}$, $i = 1, 2, \cdots, L$, sort S in ascending order to get the index sequence p_i, $i = 1, 2, \cdots, L$.

Step 2. According to the index sequence, rearrange the original i-th bit in the bit sequence as the p_i-th one in the permuted bit sequence.

With the operation of global bit permutation, the bit sequence can be transformed to a pixel plane as the encrypted image. It also can be transformed into a bit cuboid for further bit-level permutation.

3.5. The DFBC: The Proposed Image Encryption Using Hyperchaotic System, Dynamic Filtering and Bit Cuboid Operation. The framework of the DFBC is shown in Figure 6. After generating the hyperchaotic sequence, the encryption procedure consists of three steps: pixel-level diffusion (dynamic filtering and global pixel diffusion), transformation of pixel plane to bit cuboid, and bit-level permutation. The diffusion changes the gray value of every pixel while the permutation rearranges the positions of bits.

The detailed encryption procedure of the DFBC is described as follows.

Step 1. Generate the hyperchaotic sequence with initial keys by (1)-(3).

Step 2. Use the dynamic masks to diffuse the image I by the method in Section 3.2.1. Firstly, the first number of

Table 1: Testing images.

Image	Size ($X \times Y$)	Image	Size ($X \times Y$)
Lena	256×256	Cameraman	256×256
Barbara	512×512	Airfield	512×512
Histogram	512×512	Terrain	256×256
Parking	512×512	Monkey	512×512

the hyperchaotic sequence, S_0, is chosen to determine the direction of mask. Then, $X \times Y \times M$ integers are chosen from the hyperchaotic sequence. Secondly, reshape these integers into $X \times Y$ masks of $1 \times M$ or $M \times 1$, $W^{x,y}$, where the direction of mask is determined by the parity of S_0. Set the last element of the mask to one; i.e., $W_M^{x,y} = 1$. Thirdly, convolve each mask $W^{x,y}$ with its corresponding pixels to get a diffused image.

Step 3. Conduct global pixel diffusion by (17)-(18) as described in Section 3.3. The operation will expand a little change in one pixel to other pixels.

Step 4. Conduct XOR operation on the image with a hyperchaotic subsequence to change the values of pixels directly.

Step 5. Transform the pixel plane to a bit cuboid as described in Section 3.4.1. The transformed bit cuboid has as uniform width, height, and depth as possible.

Step 6. Rearrange the bits in the bit cuboid based on bit plane at height, width, and depth direction as described in Section 3.4.2.(1).

Step 7. Randomly rotate each bit-plain based on chaos sequence as described in Section 3.4.2.(3).

Step 8. Randomly symmetrize each bit-plain based on chaos sequence as described in Section 3.4.2.(2).

Step 9. Permute the bit cuboid by zigzag as described in Section 3.4.2.(4).

Step 10. Conduct global bit permutation on the bit sequence as described in Section 3.4.2.(5).

Step 11. Transform the bit sequence to a pixel plane of the original size as the final encrypted image.

The procedure of the DFBC can be categorized into five stages: hyperchaotic sequence generation (Step 1), pixel-level diffusion (Step 2-4), pixel plane to bit cuboid transformation (Step 5), bit-level permutation (Steps 6–10), and bit sequence to pixel-plane transformation (Step 11). The DFBC diffuses images with dynamic filtering, global pixel diffusion, and XOR operation on pixel-level data. And it permutes the images on bit-level data. It is worth pointing out that since the permutation on bits can change the values of pixels, the bit cuboid operations in the DFBC are capable of diffusing the images; that is to say, the bit cuboid operations have the effects of both permutation and diffusion.

The decryption process of DFBC is the inverse procedure of encryption as listed above.

4. Experimental Results

4.1. Experimental Settings. In order to evaluate the performance of the proposed DFBC, some state-of-the-art schemes, such as the hyperchaotic and DNA sequence-based method (HC-DNA) [30], the image encryption using encrypted diffusion in crisscross pattern (CDCP) [16], a class hyperchaos-based scheme (CHC) [17], an image encrypted using block-based scrambling, and image filtering (IC-BSIF) [26], are compared with some common statistical indexes. The parameters for the compared schemes are set as the corresponding papers. The parameters of the DFBC are set as follows. The initial values for the 7D hyperchaotic system are $x_1^0 = 0$, $x_2^0 = 0.1$, $x_3^0 = 0.2$, $x_4^0 = 0.3$, $x_5^0 = 0.5$, $x_6^0 = 0.6$, and $x_7^0 = 0.7$. And the preiterating times N_0 is set to 1000. The original hyperchaotic integer sequence generated from the 7D hyperchaotic equations is adopted from the beginning, which means the start position of hyperchaotic sequence is 1. Then our diffusion masks of filtering are created from this hyperchaotic integer sequence. The first byte of the hyperchaotic sequence determines the direction of 1D filter. Now that the image size is $X \times Y$ and mask size is 3×1 or 1×3, then $X \times Y \times 3$ integers of one byte are cut out from the hyperchaotic sequence to shape $X \times Y$ masks to filter every pixel $I_{x,y}$ iteratively. The round of 1D filtering and cuboid operation are both set to 1.

Six publicly accessed images and two images we collected, *Monkey* and *Parking*, with different sizes are used to test the proposed DFBC, as listed in Table 1.

The publicly accessed images *Lena, Cameraman, Barbara,* and *Airfield* are widely used in the field of image processing, including image encryption. The image *Histogram* is a graphical one which has an almost similar grayscale background and histogram which means most of its pixels are highly correlated. This image is quite different from other real photographs from cameras. The image *Terrain* is an image downloaded from the Internet which has a very single sharp mountain shape histogram. The image *Parking* is captured by the author. It has a big area of gray cloud background which has a very high correlation between pixels. The image *Monkey* is a real image captured from a video which is different from these well-known images. It is to validate that our DFBC can be applied to any real images robustly and easily.

All the experiments were conducted by Matlab 8.3 (Mathworks, Natick, MA, USA) on a 64-bit Windows 7 (Microsoft,

Redmond, WA, USA) with 8 GB memory and 3.4 GHz I3 CPU.

4.2. Security Key Analysis. A good encryption scheme should have an enough large key space and be extremely sensitive to any small changes in its security key. Both a large key space and extreme sensitivity can resist brute-force attacks. So key space and sensitivity to secret key are two essential points in encryption.

4.2.1. Key Space. Basically, the security keys of the proposed DFBC are composed of 7 initial values, i.e., $(x_1^0, x_2^0, x_3^0, x_4^0, x_5^0, x_6^0, x_7^0)$. If the precision of each initial value is 10^{-15}, the key space size is $10^{15\times7} = 10^{105} \approx 2^{348}$. From the view of cryptology, the size of the key space larger than 2^{100} can provide a high-level security [1, 47]. Therefore, the key space of the DFBC is large enough to resist all kinds of brute-force attacks from current computers. In addition, the start position of chaos sequence to form the random filter mask can also be used as a key to further enhance the key space.

4.2.2. Sensitivity to Security Key. The extreme sensitivity of an image encryption algorithm implies that even one bit changed in the keys will lead to a completely different encrypted image. In other words, if the security key changes a little, the decrypted image will be completely different from the input image.

To demonstrate the sensitivity to secret key of the DFBC, we decrypt the encrypted images twice. In the first run, we use the exact encryption keys ($x_1^0 = 0$, $x_2^0 = 0.1$, $x_3^0 = 0.2$, $x_4^0 = 0.3$, $x_5^0 = 0.5$, $x_6^0 = 0.6$, and $x_7^0 = 0.7$) to decrypt the encrypted images, while, in the second run, we attempt to decrypt the encrypted images with slightly different keys ($x_1^1 = 0 + 10^{-15}$, $x_2^1 = 0.1$, $x_3^1 = 0.2$, $x_4^1 = 0.3$, $x_5^1 = 0.5$, $x_6^1 = 0.6$, and $x_7^1 = 0.7$). We conduct the experiments on the images of Lena, Cameraman, Monkey, and Parking, and the results are shown in Figure 7. As we can see, even if we only change the key extremely little such as 10^{-15}, the decrypted images are completely different from the ones decrypted with the correct keys, validating that the proposed DFBC is highly sensitive to secret key.

4.3. Statistical Analysis. The statistical analysis is another widely used and effective way to analyze a cryptosystem. We adopt some typical statistical analysis, such as histogram analysis, information entropy, and correlation analysis, to evaluate the performance of the proposed DFBC. A cryptosystem with good performance of statistical analysis can resist all kinds of statistical attacks. The experimental analysis demonstrates that our method DFBC also can handle images especially like the graphical *Histogram* very well.

4.3.1. Histogram Analysis. Histogram is a popular and effective way to measure the distribution of pixel values in the plain image and the encrypted image for image encryption. The histogram of a plain image is usually unevenly distributed while that of an encrypted image by a good encryption scheme has a uniform distribution as much as possible. A uniform distribution of histogram indicates a totally random-like image and the least probability of recovering its plain image. In other words, in terms of the performance of encryption schemes, the flatter of the histogram of the encrypted image is, the better the encryption scheme is.

The histograms of the plain images and their corresponding encrypted images are shown in Figure 8. The histograms of the plain images are shaped like some mountains or valleys while all the histograms of encrypted images are very close to a uniform distribution. These histograms of encrypted images indicate that the proposed DFBC have the ability to resist histogram attacks. The fifth image *Histogram* especially has a highly unusual histogram, where most of the pixels have the highest gray scale levels so that the rectangles of low gray scales are very short. And our method can encrypt the image *Histogram* to be an even and flat histogram like the other photographic images. It confirms that our method works very well for man-made graphical images like *Histogram*.

The diffused images by dynamic filtering are listed in the third column of Figure 8, and the fourth column is their corresponding histograms. It is worth noting that the diffused images by dynamic filtering have almost uniformly distributed histograms like the encrypted images. It can be seen that their histograms are just a little less even than the encrypted images. Compared with the diffusion in [26], our diffusion is as good as [26] at the same level with regard to the histogram analysis. Theoretically, our 1D image filtering is more efficient and can compute faster than 2D filtering with a fixed mask.

4.3.2. Information Entropy. Information entropy (IE) is the average rate at which information is produced by a stochastic source of data. Here it is used to reflect the complexity or orderliness of the encryption system. The intensity of an 8-bit grayscale image has 2^8 possible values ($[0, 255]$). The IE is defined as

$$\mathrm{IE}(Q_i) = -\sum_{i=0}^{255} p(Q_i) \log_2 p(Q_i), \tag{20}$$

where $p(Q_i)$ is the probability that the pixel value Q_i appears [30]. When Q_i of encrypted image has the same probability, i.e., $1/256$, IE reaches maximum of 8. It matches the uniform distribution of the encrypted image perfectly.

The IEs of input images and encrypted images by different encryption methods are listed in Table 2. The third column under DF (Dynamic Filtering) is the entropies of diffusion by dynamic filtering. The fourth column under DFBC is the entropies of final encrypted images. All the entropies of diffused images except *Barbara* are improved after our bit cuboid operations, which indicates that this random permutation is very effective. It can be seen that the IEs of input images are far below 8, while those of encrypted images are very close to the ideal value. The IEs of DFBC are within $[7.9972, 7.9993]$, demonstrating that the DFBC is secure enough to resist entropy attack. Our DFBC achieves 5 out of 8 optimal values while HC-DNA and IC-BSIF achieve only 2 and 1 out of 8 optimal values, respectively. Although the IEs by DF are not the highest ones, they are very close

Figure 7: Decrypted images of Lena, Cameraman, Monkey, and Parking. The first row is with correct keys: $x_1^0 = 0$, $x_2^0 = 0.1$, $x_3^0 = 0.2$, $x_4^0 = 0.3$, $x_5^0 = 0.5$, $x_6^0 = 0.6$, and $x_7^0 = 0.7$. The second row is with slightly different keys: $x_1^1 = 0 + 10^{-15}$, $x_2^1 = 0.1$, $x_3^1 = 0.2$, $x_4^1 = 0.3$, $x_5^1 = 0.5$, $x_6^1 = 0.6$, and $x_7^1 = 0.7$.

Table 2: The IE of the testing images.

Images	Input Images	Cipher images					
		DF	DFBC	HC-DNA [30]	CDCP [16]	CHC [17]	IC-BSIF [26]
Lena	7.2283	7.9970	7.9972	7.9964	7.9968	**7.9974**	7.9973
Cameraman	7.1048	7.9950	7.9974	7.9964	**7.9976**	7.9972	7.9974
Barbara	7.6321	**7.9993**	**7.9993**	**7.9993**	7.9992	7.9992	7.9992
Airfield	7.1206	7.9991	**7.9993**	7.9992	7.9992	7.9992	**7.9993**
Histogram	4.2177	7.9893	7.9993	7.9371	7.9992	**7.9994**	7.9993
Terrain	6.5867	7.9966	**7.9974**	7.9944	7.9972	7.9970	7.9972
Parking	7.2614	7.9769	**7.9993**	**7.9993**	**7.9993**	**7.9993**	7.9992
Monkey	7.2849	7.9991	**7.9993**	7.9991	**7.9993**	**7.9993**	7.9992

to the ideal values, showing that the 1D dynamic filtering can achieve very good results of diffusion.

4.3.3. Correlation Analysis. Two neighboring pixels in a natural image usually have similar grayscale levels and thus are highly correlated. A good image encryption algorithm should result in encrypted image with extremely low correlation. Correlation coefficient γ is a widely used metric to measure the correlation and it can be formulated as [44]

$$\begin{aligned} E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)^2, \\ \operatorname{cov}(x, y) &= \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)\left(y_i - E(y)\right), \\ \gamma &= \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)\,D(y)}}, \end{aligned} \tag{21}$$

where x and y are grayscale levels of two neighboring pixels in an image and N denotes the total number of pixels in an image that satisfies $N = X \times Y$.

As listed in Table 3, we calculate the correlation coefficients for all input images and encrypted images at three different directions, i.e., horizontal γ_h, vertical γ_v, and diagonal γ_d, respectively [30]. It can be seen from this table that the correlation coefficients of all the input images are big number and close to 1 in all directions, while all the encrypted images are around 0, showing that the encryption schemes can effectively reduce the correlation of the adjacent pixels to a very low level after encryption. The DFBC outperforms the remaining schemes on 6 out of 24 correlation coefficients, whereas the CHC also achieves the optimal value 6 times and IC-BSIF achieves 5 times, showing that the DFBC has good performance for correlation of the encrypted images. When we further investigate the range of γ, however, γ of DFBC is within $[-0.0041, 0.0040]$, while those of HC-DNA, CDCP, CHC, and IC-BSIF are within $[-0.0091, 0.0076]$, $[-0.0054, 0.0048]$, $[-0.0069, 0.0048]$, and $[-0.0067, 0.0104]$, respectively. From the ranges of the algorithms, we can get the intervals of γ, i.e., 0.0081, 0.0167, 0.0102, 0.0117, and 0.0171, for

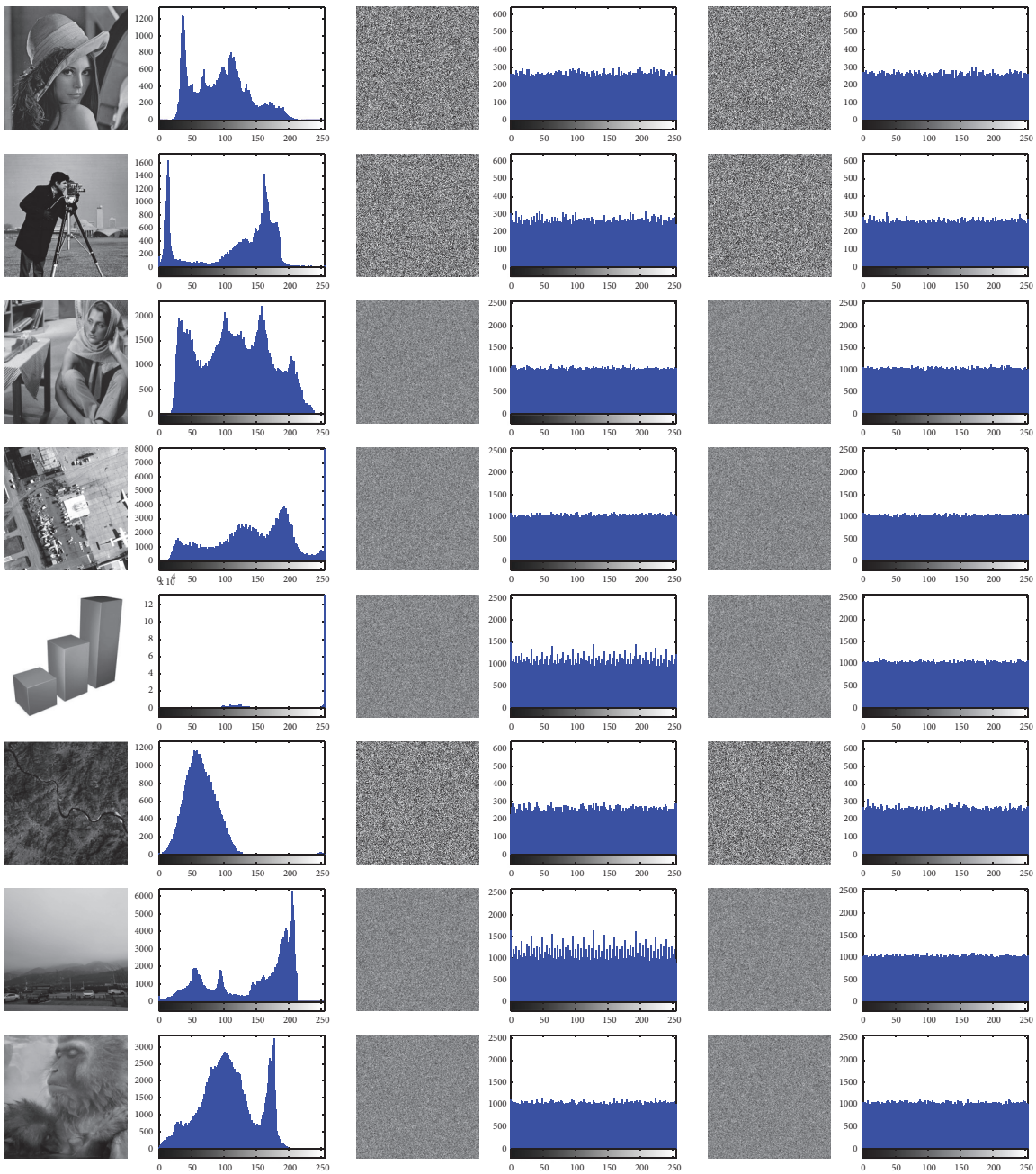


Figure 8: Histograms of the plain images and their corresponding encrypted images. The first column is plain images. The second column is the histograms of the plain images. The third column is diffusion result after 1D dynamic filtering. The fourth column is the histograms of the filtered image in the third column. The fifth column is encrypted images. And the sixth column is the histograms of the encrypted images.

DFBC, HC-DNA, CDCP, CHC, and IC-BSIF, respectively. It can be seen that the DFBC achieves the minimum interval, showing that the DFBC is the most stable one among the compared approaches regarding correlation.

To have a further correlation analysis, we randomly select 4000 pairs of neighboring pixels in horizontal direction from the input images and the encrypted images by the DFBC to show their neighboring pixel distribution maps in Figure 9. The correlation values of input images distribute near the diagonal of coordinate plane, indicating strong correlation of input images. The images *Monkey, Histogram,* and *Parking* especially have very obvious lines made by correlation points on the diagonal direction. However, the values of encrypted images distribute almost on the whole plane randomly, which shows very weak correlation of encrypted images. In other words, most of the correlation is removed by the DFBC.

4.4. Analysis of Resisting Differential Attacks. According to the theory of cryptography, an image encryption scheme should effectively resist the differential attack. Thus a good image encryption algorithm needs to be very sensitive to the plain images; that is, any trivial change (e.g., a bit or a pixel

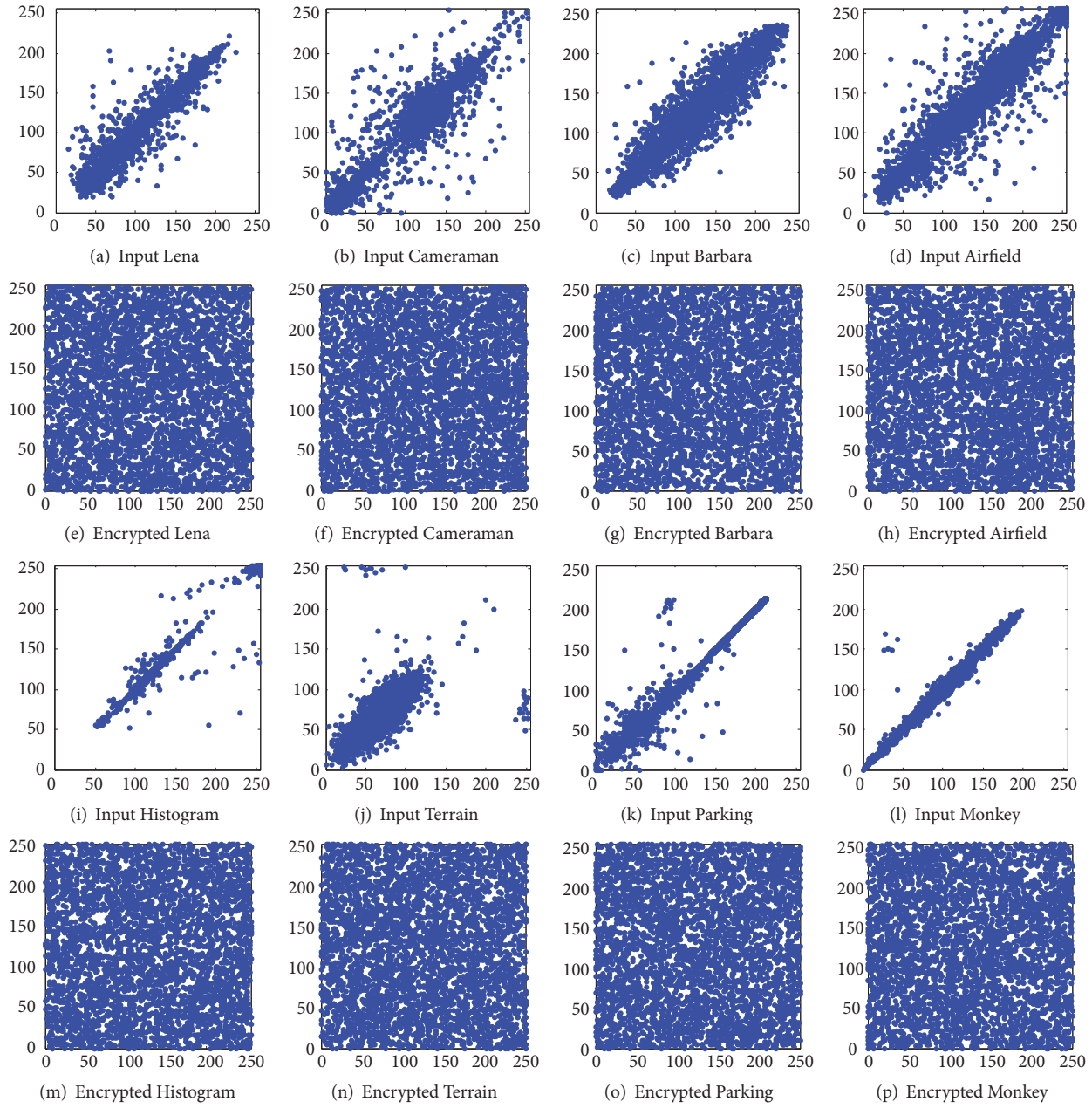


Figure 9: The adjacent-pixel distribution maps of the input images and the corresponding encrypted images in horizontal direction.

change) in a plain image can lead to a completely different encrypted image.

The number of pixels change rate (NPCR) and the unified average changing intensity (UACI) are two important metrics for differential attack analysis. The NPCR is defined as the variation ratio of two encrypted images whose plain images are slightly different, meaning the dissimilitude between two encrypted images. The UACI indicates the average intensity of the differences between the same encrypted images. In most cases, only one pixel or even one bit chosen is randomly changed to compute the NPCR and UACI. Mathematically, NPCR and UACI between two encrypted images Q^1 and Q^2 can be formulated as follows, respectively.

$$\text{NPCR} = \frac{1}{XY}\sum_{i=1}^{Y}\sum_{j=1}^{X} d_{ij} \times 100\%, \tag{22}$$

$$\text{UACI} = \frac{1}{XY}\sum_{i=1}^{Y}\sum_{j=1}^{X} \frac{\left|Q_{ij}^1 - Q_{ij}^2\right|}{255} \times 100\%, \tag{23}$$

where X and Y are the height and the width of the image, respectively, and d_{ij} is defined as

$$d_{ij} = \begin{cases} 0, & Q_{ij}^1 = Q_{ij}^2, \\ 1, & Q_{ij}^1 \neq Q_{ij}^2. \end{cases} \tag{24}$$

Table 3: The correlation coefficients γ of the testing images.

Image	γ	Input images	Cipher images				
			DFBC	HC-DNA [30]	CDCP [16]	CHC [17]	IC-BSIF [26]
Lena	γ_h	0.9494	-0.0041	0.0019	-0.0021	**0.0006**	-0.0067
	γ_v	0.9667	0.0023	-0.0030	-0.0042	**-0.0003**	0.0013
	γ_d	0.9366	0.0040	**0.0018**	-0.0022	0.0048	0.0029
Cameraman	γ_h	0.9329	-0.0023	0.0076	**-0.0022**	-0.0069	0.0104
	γ_v	0.9566	**0.0004**	-0.0091	-0.0054	-0.0044	0.0039
	γ_d	0.9117	0.0012	-0.0012	0.0048	0.0010	**0.0003**
Barbara	γ_h	0.8940	0.0027	0.0010	-0.0026	**0.0001**	0.0003
	γ_v	0.9572	-0.0029	**0.0004**	0.0006	0.0033	0.0015
	γ_d	0.8942	**-0.0005**	-0.0009	0.0005	-0.0014	0.0009
Airfield	γ_h	0.9375	**0.0003**	-0.0004	0.0010	0.0017	-0.0006
	γ_v	0.9398	0.0039	0.0002	-0.0033	-0.0003	**-0.0001**
	γ_d	0.9068	-0.0014	-0.0026	0.0013	-0.0008	**0.0004**
Histogram	γ_h	0.9933	**0.0004**	-0.0012	-0.0016	-0.0012	0.0040
	γ_v	0.9947	0.0013	**0.0008**	-0.0009	0.0042	0.0016
	γ_d	0.9889	0.0011	0.0006	**0.0001**	-0.0004	0.0005
Terrain	γ_h	0.8703	0.0030	-0.0049	**-0.0005**	0.0048	-0.0017
	γ_v	0.6673	-0.0026	-0.0028	-0.0024	0.0011	**0.0004**
	γ_d	0.6601	0.0012	-0.0012	-0.0019	**0.0005**	0.0017
Parking	γ_h	0.9957	-0.0014	**-0.0002**	0.0007	-0.0004	-0.0020
	γ_v	0.9888	0.0006	-0.0041	0.0018	0.0034	**-0.0005**
	γ_d	0.9896	**0.0004**	0.0029	0.0004	-0.0006	0.0014
Monkey	γ_h	0.9955	**-0.0002**	0.0013	-0.0021	0.0032	-0.0046
	γ_v	0.9912	-0.0012	-0.0016	-0.0039	**0.0010**	-0.0010
	γ_d	0.9938	0.0004	0.0007	-0.0008	**-0.0000**	0.0007

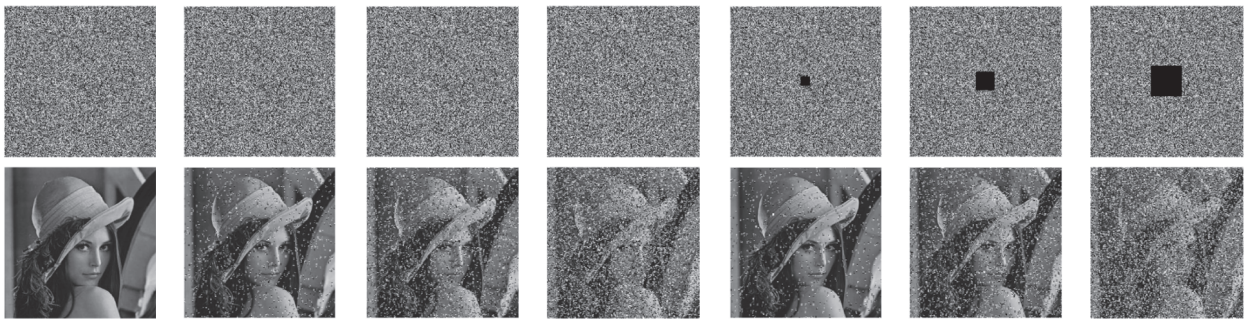
NPCR concentrates on the absolute number of pixels which change values in differential attack, while the UACI focuses on the average difference between the corresponding encrypted images. For an image with 256 grayscale levels, the expectations of NPCR and UACI of an ideally encrypted image are 99.6094% and 33.4635%, respectively [48]. Generally speaking, the more NPCR gets close to 100% and the bigger UACI is, the more effective it is for the encryption scheme to resist differential attacks.

We randomly change one pixel and then add 1 on gray value in the plain images to compute one value of NPCR and UACI. We repeat the process 10 times and the average scores of NPCR and UACI of the corresponding image encryption schemes are recorded in Tables 4 and 5, respectively, as the final NPCR and UACI.

In Table 4, although the NPCR scores by the DFBC are not as good as those by CDCP, they are very close to the maximum theoretical scores. DFBC apparently outperforms HC-DNA in terms of NPCR, and it achieves comparable results with CHC and IC-BSIF. In Table 5, DFBC obviously outperform all the other schemes in 6 out of 8 cases, whereas HC-DNA still gets the poorest results in all cases. The values of NPCR and UACI indicate that the DFBC is able to resist differential attacks.

4.5. Robustness Analysis. A digital image is inevitably contaminated by noise or has data loss during storage and transmission. Image encryption should have the ability to resist both noise and data loss. The experimental results of the DFBC for robustness analysis are shown in Figure 10. We firstly add 0.5%, 1%, and 2% salt and pepper noise to the encrypted Lena, and the corresponding decrypted images are shown in Figures 10(b)–10(d), respectively. It can be seen that, for 0.5% and 1% salt and pepper noise, although the decrypted image contains noise, it can clearly recover the original image very well. However, when the noise increases to 2%, we can only see the outline of Lena. When the encrypted image has 0.4% and 1.56% data loss, the proposed DFBC can recover Lena, as shown in Figures 10(e)-10(f), respectively. For 4% data loss, the decrypted image retains some information for us to recognize Lena, as shown in Figure 10(g).

From the analysis, we can see that the proposed DFBC is robust to a certain extent. However, as pointed out by Hua and Zhou, resisting noise, rotation, and cropping attacks are



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

Figure 10: Robustness analysis results of noise and data loss. (a) The encrypted Lena and its decrypted image. (b)-(d) The encrypted Lena with 0.5%, 1%, and 2% salt and pepper noise and its decrypted image, respectively. (e)-(g) The encrypted Lena with 0.4%, 1.56% and 4% data loss and its decrypted image, respectively.

Table 4: The average NPCR (%) of running the schemes 10 times.

Image	DFBC	HC-DNA [30]	CDCP [16]	CHC [17]	IC-BSIF [26]
Lena	99.6234	54.7104	**100.0000**	99.6135	99.6213
Cameraman	99.6338	52.8110	**100.0000**	99.6069	99.6085
Barbara	99.5835	44.7173	**99.6531**	99.6124	99.6095
Airfield	99.6086	46.5910	99.5702	99.6043	**99.6117**
Histogram	99.5979	63.8850	**99.6847**	99.6037	99.6064
Terrain	99.6111	40.6036	**100.0000**	99.6268	99.6121
Parking	99.6087	57.5092	99.5401	**99.6143**	99.6114
Monkey	99.6201	47.8650	**99.6702**	99.6037	99.6154

Table 5: The average UACI (%) of running the schemes 10 times.

Image	DFBC	HC-DNA [30]	CDCP [16]	CHC [17]	IC-BSIF [26]
Lena	33.5092	22.5711	**33.5558**	33.4361	33.4072
Cameraman	**33.5578**	19.9770	33.4968	33.5085	33.4937
Barbara	33.4808	18.0029	33.4835	**33.5076**	33.5071
Airfield	**33.4760**	19.5391	33.4363	33.4614	33.4667
Histogram	**33.4866**	26.4724	33.4677	33.4690	33.4620
Terrain	**33.5198**	18.9648	33.4829	33.4565	33.4323
Parking	**33.5882**	23.9542	33.4643	33.4941	33.4453
Monkey	**33.5031**	20.5115	33.4659	33.4692	33.4536

current limitations for image encryption by filtering [26]. The proposed DFBC will be significantly improved when such limitations are resolved.

5. Conclusions

Image security is one of the most important branches of information security. To enhance image security, in this paper, we have proposed a novel image encryption algorithm that combines a 7D hyperchaotic system with 5 positive LEs, dynamic filtering operation, and bit cuboid operations, namely, DFBC, for image encryption. Extensive experiments have shown that the proposed DFBC can outperform some state-of-the-art image encryption schemes in terms of several evaluation criteria, indicating that the DFBC is effective for image encryption. In the future, we will study construction of more complex key for the DFBC. Furthermore, since the dynamic filtering can be performed in parallel, we will also implement the algorithm on the framework of CUDA and use GPU to accelerate it.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported partly by the Fundamental Research Funds for the Central Universities (Grants no. JBK1902029, no. JBK1802073, and no. JBK170505), Sichuan Science and Technology Program (Grant no. 19ZDYF0040), the Natural Science Foundation of China (Grant no. 71473201), and the Scientific Research Fund of Sichuan Provincial Education Department (Grant no. 17ZB0433).

References

[1] B. Schneier, *Applied Cryptography: Protocols, Algorithms and Source Code in C*, John Wiley & Sons, Indianapolis, IN, USA, 2007.

[2] Taiyong Li, Minggao Yang, Jiang Wu, and Xin Jing, "A Novel Image Encryption Algorithm Based on a Fractional-Order Hyperchaotic System and DNA Computing," *Complexity*, vol. 2017, Article ID 9010251, 13 pages, 2017.

[3] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[4] Y. Wang, K.-W. Wong, X. Liao, and G. Chen, "A new chaos-based fast image encryption algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 514–522, 2011.

[5] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Processing*, vol. 92, no. 4, pp. 1101–1108, 2012.

[6] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-Box," *Information Sciences*, vol. 450, pp. 361–377, 2018.

[7] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 305–313, 2018.

[8] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image and Vision Computing*, vol. 24, no. 9, pp. 926–934, 2006.

[9] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D Logistic-Sine-coupling map for image encryption," *Signal Processing*, vol. 149, pp. 148–161, 2018.

[10] D. Arroyo, R. Rhouma, G. Alvarez, S. Li, and V. Fernandez, "On the security of a new image encryption scheme based on chaotic map lattices," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 18, no. 3, Article ID 033112, 2008.

[11] M. L. Sahari and I. Boukemara, "A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption," *Nonlinear Dynamics*, vol. 94, no. 1, pp. 723–744, 2018.

[12] Z. Nanrun, Y. Xingyu, L. Haoran, T. Xiangyang, and L. Guangyong, "Multi-image encryption scheme based on quantum 3d arnold transform and scaled zhongtang chaotic system," *Quantum Information Processing*, vol. 17, no. 12, p. 338, Oct 2018.

[13] M. T. Rosenstein, J. J. Collins, and C. J. de Luca, "A practical method for calculating largest Lyapunov exponents from small data sets," *Physica D: Nonlinear Phenomena*, vol. 65, no. 1-2, pp. 117–134, 1993.

[14] A. Wolf, J. B. Swift, and H. L. a. Swinney, "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

[15] J. Wang and G.-P. Jiang, "Cryptanalysis of a hyper-chaotic image encryption algorithm and its improved version," *Acta Physica Sinica*, vol. 60, no. 6, 2011.

[16] C.-X. Zhu, Y.-P. Hu, and K.-H. Sun, "New image encryption algorithm based on hyperchaotic system and ciphertext diffusion in crisscross pattern," *Journal of Electronics & Information Technology*, vol. 34, no. 7, pp. 1735–1743, 2012.

[17] C.-X. Zhu and K.-H. Sun, "Cryptanalysis and improvement of a class of hyperchaos based image encryption algorithms," *Acta Physica Sinica*, vol. 61, no. 12, p. 120503, 2012.

[18] X.-L. Chai, Z.-H. Gan, Y. Lu, M.-H. Zhang, and Y.-R. Chen, "A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system," *Chinese Physics B*, vol. 25, no. 10, 2016.

[19] Q. Ran, L. Wang, J. Ma, L. Tan, and S. Yu, "A quantum color image encryption scheme based on coupled hyper-chaotic Lorenz system with three impulse injections," *Quantum Information Processing*, vol. 17, no. 8, Art. 188, 30 pages, 2018.

[20] H. Zhu, X. Zhang, H. Yu, C. Zhao, and Z. Zhu, "An image encryption algorithm based on compound homogeneous hyper-chaotic system," *Nonlinear Dynamics*, vol. 89, no. 1, pp. 61–79, 2017.

[21] H. Xue, J. Du, S. Li, and W. Ma, "Region of interest encryption for color images based on a hyperchaotic system with three positive Lyapunov exponets," *Optics & Laser Technology*, vol. 106, pp. 506–516, 2018.

[22] X. Li, W. Chen, and Y. Wang, "Quantum Image Compression-Encryption Scheme Based on Quantum Discrete Cosine Transform," *International Journal of Theoretical Physics*, vol. 57, no. 9, pp. 2904–2919, 2018.

[23] X. Wu, D. Wang, J. Kurths, and H. Kan, "A novel lossless color image encryption scheme using 2D DWT and 6D hyperchaotic system," *Information Sciences*, vol. 349-350, pp. 137–153, 2016.

[24] X. Chai, Z. Gan, and M. Zhang, "A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion," *Multimedia Tools and Applications*, vol. 76, no. 14, pp. 15561–15585, 2017.

[25] L. Xu, X. Gou, Z. Li, and J. Li, "A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion," *Optics and Lasers in Engineering*, vol. 91, pp. 41–52, 2017.

[26] Z. Hua and Y. Zhou, "Design of image cipher using block-based scrambling and image filtering," *Information Sciences*, vol. 396, pp. 97–113, 2017.

[27] D. Liu, W. Zhang, H. Yu, and Z. Zhu, "An image encryption scheme using self-adaptive selective permutation and inter-intra-block feedback diffusion," *Signal Processing*, vol. 151, pp. 130–143, 2018.

[28] G. Ye, C. Pan, X. Huang, and Q. Mei, "An efficient pixel-level chaotic image encryption algorithm," *Nonlinear Dynamics*, vol. 94, no. 1, pp. 745–756, 2018.

[29] Y. Xiong, C. Quan, and C. J. Tay, "Multiple image encryption scheme based on pixel exchange operation and vector decomposition," *Optics and Lasers in Engineering*, vol. 101, pp. 113–121, 2018.

[30] K. Zhan, D. Wei, J. Shi, and J. Yu, "Cross-utilizing hyperchaotic and DNA sequences for image encryption," *Journal of Electronic Imaging*, vol. 26, no. 1, Article ID 013021, 2017.

[31] S. Sun, "A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling," *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–14, 2018.

[32] X. Chai, Z. Gan, K. Yang, Y. Chen, and X. Liu, "An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations," *Signal Processing: Image Communication*, vol. 52, pp. 6–19, 2017.

[33] A. ur-Rehman, X. Liao, M. A. Hahsmi, and R. Haider, "An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using DNA and chaos," *Optik - International Journal for Light and Electron Optics*, vol. 153, pp. 117–134, 2018.

[34] S. Cai, L. Huang, X. Chen, and X. Xiong, "A Symmetric Plaintext-Related Color Image Encryption System Based on Bit Permutation," *Entropy*, vol. 20, no. 4, p. 282, 2018.

[35] N. Zhou, W. Chen, X. Yan, and Y. Wang, "Bit-level quantum color image encryption scheme with quantum cross-exchange operation and hyper-chaotic system," *Quantum Information Processing*, vol. 17, no. 6, Art. 137, 24 pages, 2018.

[36] L. Teng, X. Wang, and J. Meng, "A chaotic color image encryption using integrated bit-level permutation," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 6883–6896, 2018.

[37] Y. Dai, H. Wang, and Y. Wang, "Chaotic Medical Image Encryption Algorithm Based on Bit-Plane Decomposition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 4, 2016.

[38] A. Shafique and J. Shahid, "Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps," *The European Physical Journal Plus*, vol. 133, no. 8, 2018.

[39] Q. Yang, D. Zhu, and L. Yang, "A new 7D hyperchaotic system with five positive Lyapunov exponents coined," *International Journal of Bifurcation and Chaos*, vol. 28, no. 5, 1850057, 20 pages, 2018.

[40] M. Brindha and N. Ammasai Gounden, "A chaos based image encryption and lossless compression algorithm using hash table and Chinese Remainder Theorem," *Applied Soft Computing*, vol. 40, pp. 379–390, 2016.

[41] X. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dynamics*, vol. 62, no. 3, pp. 615–621, 2010.

[42] S. Sheng, X. Zhang, and G. Lu, "Finite-time outer-synchronization for complex networks with Markov jump topology via hybrid control and its application to image encryption," *Journal of The Franklin Institute*, vol. 355, no. 14, pp. 6493–6519, 2018.

[43] Q. Wang, M. Wei, X. Chen, and Z. Miao, "Joint encryption and compression of 3D images based on tensor compressive sensing with non-autonomous 3D chaotic system," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 1715–1734, 2018.

[44] Z. Wang, X. Huang, Y.-X. Li, and X.-N. Song, "A new image encryption algorithm based on the fractional-order hyperchaotic Lorenz system," *Chinese Physics B*, vol. 22, no. 1, Article ID 010504, 2013.

[45] X. Lv, X. Liao, and B. Yang, "Bit-level plane image encryption based on coupled map lattice with time-varying delay," *Modern Physics Letters B. Condensed Matter Physics, Statistical Physics, Applied Physics*, vol. 32, no. 10, 1850124, 25 pages, 2018.

[46] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Processing*, vol. 148, pp. 124–144, 2018.

[47] D. R. Stinson, *Cryptography: Theory and Practice*, CRC press, 2005.

[48] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber Journals: Multidisciplinary Journals in Science and Technology*, vol. 2, pp. 31–38, 2011.