applied sciences

MDPI

Article

Joint Image Compression and Encryption Using IWT with SPIHT, Kd-Tree and Chaotic Maps

Nasrullah [1,2], **Jun Sang** [1,2,*], **Muhammad Azeem Akbar** [1,2], **Bin Cai** [1,2], **Hong Xiang** [1,2] **and Haibo Hu** [1,2]

[1] Key Laboratory of Dependable Service Computing in Cyber Physical Society of Ministry of Education, Chongqing University, Chongqing 400044, China; Nasrullah.ce@cqu.edu.cn (N.); azeem.akbar@ymail.com (M.A.A.); caibin@cqu.edu.cn (B.C.); xianghong@cqu.edu.cn (H.X.); hbhu@cqu.edu.cn (H.H.)

[2] School of Big Data & Software Engineering, Chongqing University, Chongqing 401331, China

* Correspondence: jsang@cqu.edu.cn

Received: 11 September 2018; Accepted: 15 October 2018; Published: 17 October 2018

Abstract: Confidentiality and efficient bandwidth utilization require a combination of compression and encryption of digital images. In this paper, a new method for joint image compression and encryption based on set partitioning in hierarchical trees (SPIHT) with optimized Kd-tree and multiple chaotic maps was proposed. First, the lossless compression and encryption of the original images were performed based on integer wavelet transform (IWT) with SPIHT. Wavelet coefficients undergo diffusions and permutations before encoded through SPIHT. Second, maximum confusion, diffusion and compression of the SPIHT output were performed via the modified Kd-tree, wavelet tree and Huffman coding. Finally, the compressed output was further encrypted with varying parameter logistic maps and modified quadratic chaotic maps. The performance of the proposed technique was evaluated through compression ratio (CR) and peak-signal-to-noise ratio (PSNR), key space and histogram analyses. Moreover, this scheme passes several security tests, such as sensitivity, entropy and differential analysis tests. According to the theoretical analysis and experimental results, the proposed method is more secure and decreases the redundant information of the image more than the existing techniques for hybrid compression and encryption.

Keywords: chaotic maps; encryption; image compression; integer wavelet transform; k-dimensional tree; set partition in hierarchical trees

1. Introduction

In the current era, with the availability of inexpensive capturing devices with high resolution, the rapid growth of image transmission over public networks has raised substantial concern in the fields of secure transmission and the compression of images [1,2].

The compression of data is the main thrust in multimedia applications due to high redundancy. Without compression, storage and communication resources are used inefficiently. In particular, low-bandwidth communication channels and power-limited devices require low-bit-rate compression due to bandwidth and power constraints.

In addition, encryption is required for confidentiality and integrity of the information, particularly for open-access networks. Encryption algorithms are typically optimized for a specific type of data due to several factors, such as encryption speed, processing power and sensitivity to bit changes. Encryption algorithms that are specially designed for text data encryptions, such as data encryption standard (DES), advance encryption standard (AES), and Blowfish, are not suitable for image data encryption. Image data have various built-in features, such as redundancy and strong correlation among adjacent pixels. Therefore, specific encryption techniques are required for such a bulky data. In comparison to traditional encryption techniques, chaotic map cryptosystems are considered more suitable for image encryption [3–12]. However, some chaotic cryptosystems have well-known weaknesses such as relatively small key size, short-length chaotic orbits and lack of the confusion property [4,13].

A joint compression and encryption scheme are a promising method for transmitting image data securely and efficiently over public networks. By combining the compression and encryption operations, more uncertainty is achieved in the size of the encrypted image. Due to this uncertainty, the difficulty level that would be faced by attackers is increased. Currently, there exist three approaches in this field: The first method is to achieve compression via encryption algorithms [1]. In this method, compression capability is embedded in an encryption algorithm. However, this method is not suitable for image data because image compression capability is compromised. The second method is based on a compression algorithm and encryption is an additional capability that is realized via entropy coding techniques [1,2]. However, this method is based on a compression algorithm and encryption is embedded as an additional functionality. Therefore, the security levels are not appropriate. The third scheme is based on compression and encryption serially [1,14–16]. Unlike the former two methods, the compression and encryption serially are considered equally important. However, by separating these two operations, we cannot achieve high uncertainty in the length of the cipher image.

In this paper, a joint image compression and encryption scheme that is based on IWT with SPIHT, Kd-tree and multiple chaotic maps is proposed. The main contribution of this work is to use high dimensional kd-tree for the diffusion and permutations of integer wavelet coefficients and SPIHT output in such a way that maximum compression is achieved. In the proposed three stage scheme, joint compression and encryption was performed at first two stages. At third stage only encryption was performed through multiple chaotic maps. The proposed method provides high security and much more compression than the existing joint image compression and encryption techniques. The experimental results that are presented in this paper demonstrate the effectiveness of the proposed joint image encryption and compression scheme. This research evaluates the Kd-tree compact data structure for permutation and substitution of wavelet coefficients and SPIHT bit stream for the compression and encryption simultaneously. Kd-tree is used here to find the homogenous regions of zeros and ones in the bit stream for the further compression of SPIHT output.

The remainder of this paper is organized as follows: In Section 2, we discuss related works. A review of Kd-tree and wavelet tree for compact data structures is given in Section 3. In Section 4, the proposed image compression and encryption scheme is described. The experimental results and security analysis of the proposed scheme are presented in Section 5. Finally, the conclusions are presented in Section 6.

2. Related Works

There exist many image compression and encryption algorithms. However, to achieve robust security and compression together is still a challenging task because both are contradictory things. In this section, we will discuss SPIHT based image compressions, chaotic map-based image encryptions and some joint image compression and encryption schemes.

2.1. SPIHT-Based Image Compression

SPIHT is a wavelet-based coding method that was proposed by Said and Pearlman in 1996 [17]. Wavelet transform provides satisfactory time and frequency resolution simultaneously. Thus, this method is considered highly suitable for the analysis and compression of images. However, the high computational cost of the traditional wavelet transform limits its application in high-speed and real-time signal processing. In 1998, Sweldens proposed the Lifting wavelet transform [18], which is an alternative solution for discrete wavelet transform (DWT). Compared with the traditional wavelet transform, the lifting algorithm does not depend on the Fourier transform and requires less computations (up to 50%) compared to the convolution-based approach. The Lifting wavelet transform is highly suitable for implementation on the hardware and reduces the computational complexity. During the lifting implementation, no extra memory buffer is required because of the in-place computation feature of lifting. Lifting wavelet transform (LWT) has attracted increased interest in image compression as it has low computational complexity due to the utilization of an ordinary wavelet filter in the lifting steps [18]. The lifting-based approach offers an integer-to-integer transformation that is suitable for lossless image compression.

LWT is derived from a polyphase matrix representation that can distinguish between even and odd samples. A polyphase matrix is a matrix in which the elements are filter masks. To derive the lifting wavelet transform, the original filter is divided into a series of shorter filters using the filter-factoring algorithm. Those filters are designed as low-pass and high-pass filters with lifting steps. A polyphase matrix is used for matrix decomposition, which leads to additional matrices in the form of a lifting scheme.

There are three stages in the LWT (see Figure 1): splitting, prediction and update.

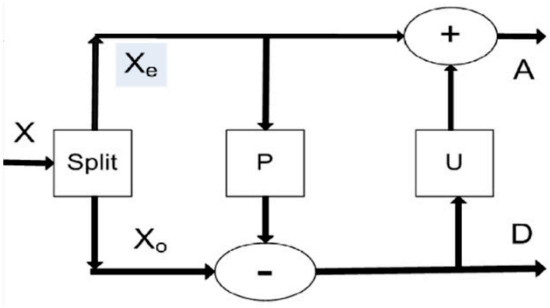


Figure 1. Single lifting step.

In the splitting stage, the input signal is divided into two sets: odd (Equation (1)) and even (Equation (2)) signals, which can be described as:

$$x_o(n) = x(2n+1) \tag{1}$$

$$x_e(n) = x(n+1) \tag{2}$$

In the prediction stage, the even signal is used to estimate the odd signal via multiplication by a forecasting parameter P. The difference between the odd and predicted signals is the high-frequency coefficients, as expressed in Equation (3).

$$d(n) = x_o(n) - p(x_e(n)) \tag{3}$$

Then, in the update stage, the even signal is updated, and it results in low-frequency coefficients by Equation (4).

$$c(n) = x_e(n) - U(d(n)) \tag{4}$$

SPIHT is one of the most widely used, powerful and efficient image compression techniques, with extremely low computational complexity, especially with lifting-based IWT. With SPIHT, the image is first decomposed into a series of wavelet coefficients via the wavelet transform. These wavelet coefficients are grouped into sets that are known as spatial orientation trees (SOT), as shown in Figure 2a and scanning sequence in Figure 2b. After that, the coefficients in each SOT are encoded progressively from the most significant bit planes to the least significant bit planes, starting with the coefficients with the highest magnitude.

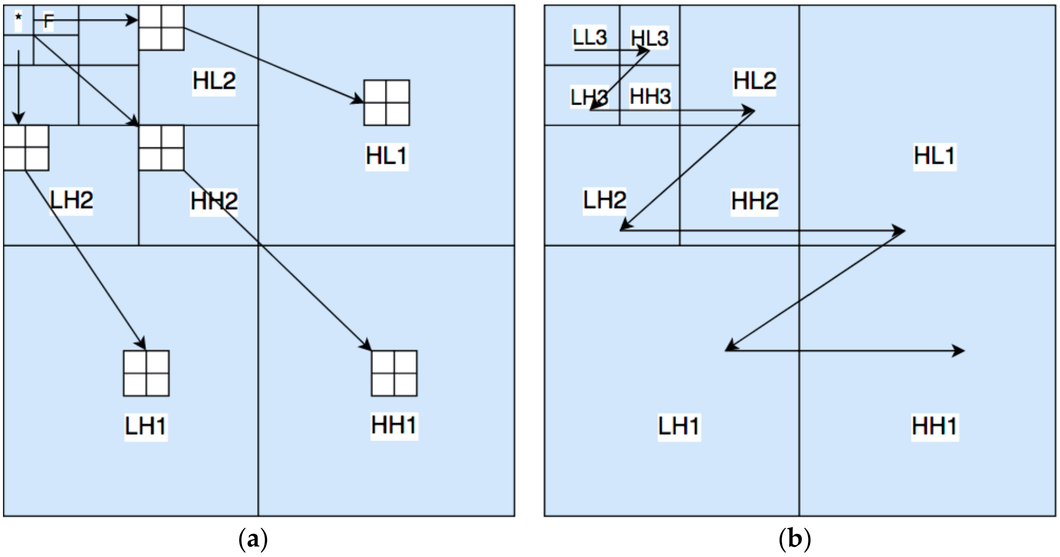


Figure 2. (**a**) SOT parent-child relationship (**b**). Scanning sequence of wavelet coefficients in SPIHT.

The SPIHT algorithm involves two coding passes: the sorting pass and the refinement pass. The sorting pass looks for zero trees and sorts significant and insignificant coefficients with respect to a specified threshold, as shown in Equation (5). The refinement pass sends the precision bits of the significant coefficients. After one sorting pass and one refinement pass, which can be considered one scan pass, the threshold T is halved, and the coding process is repeated until the expected bit rate is achieved. At the end, the compressed bit stream for all the coefficients is obtained. It provides satisfactory image quality, high PSNR, and optimization for progressive image transmission. SPIHT is highly vulnerable to bit corruption since a single bit error can introduce major image distortion, depending on its location.

$$S_n(T) = \begin{cases} 1 & \max_{(i,j)\epsilon T}\{[|c_{i,j}|] \geq 2^n \\ 0 & otherwise \end{cases} \tag{5}$$

2.2. Image Encryption with Chaotic Maps

Many encryption algorithms, specifically for digital images, have been proposed to meet the requirements of robust image encryption applications. Among them, the chaos-based encryption algorithms are popular due to several excellent properties, such as ergodicity, randomness, unpredictability, and high sensitivity to initial conditions and various parameters [5–7].

However, most of these schemes encounter various problems that limit their robustness and security [19–24]. It is well known that some chaotic maps completely lose their chaotic behavior and become periodic when they are discretized [25]. In addition, the security of chaotic systems depends on control parameters and shows specific behavior on these values, which can be determined in some cases. Therefore, one of the main problems in chaos-based cryptography is the selection of the chaotic system [21].

Various types of chaotic maps have been proposed, such as the Logistic map, Sine map, Cubic map, Tent map, Baker map, Arnold map, Chen map and many more. Among these, the most primitive and complex is the logistic map, which arises from a nonlinear dynamical equation [24]. However, all these maps suffer from security deficiencies. The security of chaotic maps depends upon the number of parameters that are used in these chaotic cryptosystems. To increase the number of these parameters, multi-dimensional and multiple iterative chaotic maps have been used [26]. However, security deficiencies remain [27]. The control parameters in chaotic systems act as a key on which security of the system is based. It is well known that the secret key is of fundamental importance in the security of all cryptosystems. There exist several tools and methods for estimating the initial values of chaotic systems by determining the behavior of the systems [28]. Therefore, some researchers used chaotic and non-chaotic encryption schemes together [29].

Recently, another type of chaotic system was reported, namely, the hyper-chaotic system, which has superior properties compared to general chaotic systems. It has two or more Lyapunov exponents and more-complex structures. These characteristics are considered to provide a strong entropy source. However, cryptanalysis studies demonstrate that hyper chaotic schemes can be broken by chosen-plaintext attack; thus, these are also insecure and not suitable for secure image communication [30].

For a fixed chaotic system, with the development of chaos theory, the chaotic orbits may be estimated, and their parameters or initial values may be predicted [31]. Therefore, it is not secure to use a fixed chaotic system in the encryption algorithm. Compared with fixed chaotic maps, time-varying chaotic maps are more complex and the output sequences are non-stationary, which makes them much more difficult to predict and analyze [31]. The varying parameters disrupt the phase space of the system; hence, the system can resist phase-space reconstruction attacks and chaotic signal estimation technologies effectively [32]. Thus, an image encryption algorithm that is based on a time-varying chaotic system is more secure. A direct and effective method for constructing a time-varying chaotic system is through varying parameters of the chaotic system. Recently, several chaotic image encryption algorithms that are based on varying parameters have been proposed [31,32]. In [32], a parameter-varying logistic map, which can overcome the weaknesses of the logistic map and resist phase-space reconstruction attacks, is used to shuffle the plain image.

2.3. Joint Image Compression and Encryption

Several joint image compression and encryption algorithms have been proposed in the literature that achieves robust security in the compressed domain. Maqbool et al. [33] proposed a secure JPEG algorithm for simultaneous compression and encryption. During JPEG compression, encryption is introduced by scrambling the DCT coefficients in the quantization step. Xiaoyong et al. [34] proposed a hybrid compression and encryption algorithm that is based on the generalized knight's tour, DCT and the Chen chaotic map. In this method, the nested generalized knight's tour was used to scramble the plain image, and DCT and quantization coding were utilized to compress the image with a high compression ratio. Diffusion was performed by encrypting some of the DCT coefficients by using the Chen chaotic system. Tong et al. [35] proposed joint compression and encryption using integer wavelet transform, SPIHT and multiple chaotic maps. First, integer wavelet coefficients are scrambled with the Lorenz map and the Henon map and compressed using SPIHT. At the end, the SPIHT bit stream is encrypted with a logistic map sequence. Zhang et al. [36] proposed a hybrid technique by combining compression using SPIHT and a chaos-based cryptosystem to yield a crypto-compression scheme that is based on a modified logistic map. In [37], joint compression and encryption of image data through cross-coupled chaotic maps and Non-uniform Discrete Cosine Transform was proposed. The compression of data is performed on blocks and permutation of these blocks and diffusion are carried out simultaneously. In [14], a joint lossless image encryption and compression scheme that is based on integer wavelet transform (IWT), set partitioning in hierarchical trees (SPIHT) with a hyper-chaotic system, a nonlinear inverse operation, and Secure Hash Algorithm-256 (SHA-256) was proposed. A JPEG based compression encryption algorithm is presented in [38]. This scheme is known as content adaptive because of the encryption key for altering the discrete cosine transform (DCT) orthogonal transformation and the encryption of the AC and DC coefficients performed through the key generated via BLAKE2 hashing algorithm applying to the plain image.

3. Compact Data Structures

Compact data structures offer a fascinating approach in substitution, data compression and efficient retrieval of information. In this paper, we used two well-known compact data structures methods: Kd-tree and wavelet tree for the compression, substitution and permutations of the SPIHT bit stream.

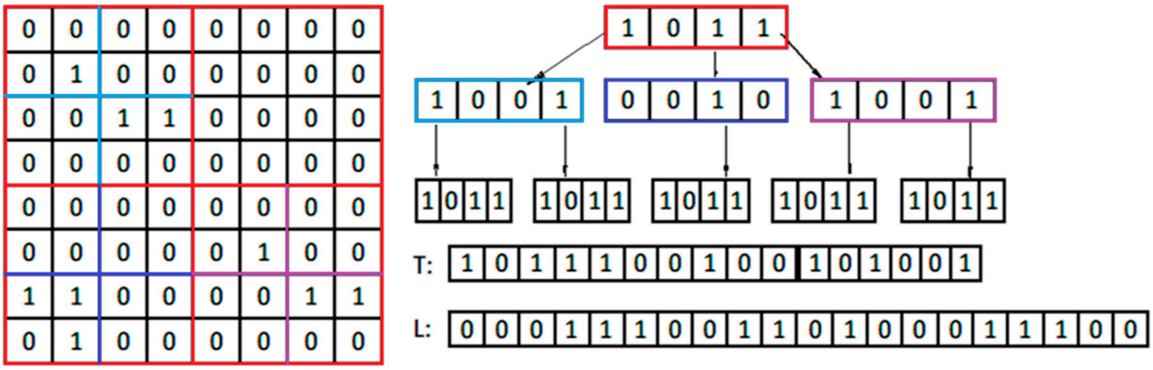
3.1. Kd-Tree

Initially, Kd-tree was used for the representation of web graphs. Here, we used Kd-tree for the compression of binary sparse matrices with permutations and substitution operations. The Kd-tree structure uses various matrix properties, such as sparseness and large homogeneous areas of zeros and ones. Kd-tree efficiently compresses the large areas of zeros and ones and provides efficient navigation over these compressed values. The simplest variant of Kd-tree is K^2-tree, where; $K = 2$; K^2-tree is similar to a compact Quad tree [39].

In K^2-tree, a binary matrix is divided recursively from left to right and from top to bottom. A conceptual tree is built from this recursive division; a 1-bit is added to the sub-tree T (root tree) of the conceptual tree if the sub-matrix contains at least a single 1-bit and 0-bits otherwise. At each partitioning level, each matrix is divided into K^2 sub-matrices of equal size. During this recursive division of the binary matrix, two sub-trees (T and L) are built:

- T (tree): stores all the level-wise traversal bits, except the last-level values when the original matrix values have been reached. The bits in the T tree are placed level-wise.
- L (last-level leaves): stores leave values of the original binary matrix after recursive subdivisions with replacement of consecutive zeros and ones (more than three).

Figure 3 shows an example of a K^2-tree in which the matrix was divided into four sub-matrices and for each sub-matrix, the root node has a bit. Each bit indicates whether the sub-matrix has at least one 1 or all zeros. Therefore, the root node of Figure 3 is 1011, since the first sub-matrix has 1 s, the second does not have 1 s, and so on. The sub-matrices that have 1 s are divided again, following the same method, until the subdivision reaches the cell level.



0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0
1	1	0	0	0	0	1	1
0	1	0	0	0	0	0	0

1 0 1 1
1 0 0 1
0 0 1 0
1 0 0 1
1 0 1 1
1 0 1 1
1 0 1 1
1 0 1 1
1 0 1 1
T: 1 0 1 1 1 0 0 1 0 0 1 0 1 0 0 1
L: 0 0 0 1 1 1 0 0 1 1 0 1 0 0 0 1 1 1 0 0

Figure 3. Binary matrix with K^2-tree example where K = 2.

The original binary matrix can be constructed by navigating these trees. For this, we must construct a rank structure with sub-tree T. The rank structure counts the number of ones in the root sub-tree T. Each 1-bit in T represents a sub-matrix, depending upon its position, and each 0-bit represents a zero matrix. If we know the position of slightly in root tree T, we can determine the position of its children via pos0 = rank1 (T, pos) × K^2 of T: L. In this way, navigate the conceptual tree using trees T and L.

Variants of the K^2-tree can be used to compress contiguous zeros and ones, or any other specific pattern, together. With these variants of kd-tree, we can compress binary images that contain large areas of homogeneous values. In these variants, the strategy is to check for specific contiguous values and remove these contagious values from the original matrix via a specific bit representation in the root tree. The sizes of these homogeneous areas vary with the level. If we remove homogenous regions of multiple values from the original matrix, then we need more bits in the root tree to represent the values in the original matrix. In our case, we compress homogenous regions of zeros and ones. Conceptually, these values are represented by nodes of different colors. The variants of the K^2-tree for compression of zeros and ones can be traversed in the same way as the original K^2-tree. With modified navigation rules, we can easily implement the compression of ones with K^2-trees.

Depending upon the size and distribution of the values in the binary matrix, we can change the number of division levels by changing the value of K. Usually, for a very large matrix, higher values of K at the start and smaller values at lower levels improve compression. To further improve the compression efficiency, we have used different values of K for ones and zeros, which also improve the security level by introducing more uncertainty in the data size. These variants of K^2-tree can be used efficiently for the compression of binary raster images.

3.2. Wavelet Tree

The wavelet tree is a very useful data structure that serves several purposes. It can be used to adaptively change the entropy measures of the data when it encodes, thereby enabling compressed data representations. It is considered useful for three tasks: (i) adaptive representation of a sequence, (ii) reordering of data elements, and (iii) for the grid of a point's representation [40].

A wavelet tree is constructed by assigning 0 to half of the symbol and 1 to the other half of the symbol. The same process for symbol assignment is performed recursively on each half of the children, splitting the αbets at each level. The splitting and symbol assignment process stops when each 03B1bet set consists of only two symbols.

In Figure 4, we demonstrate a simple wavelet tree as an example. The representation shows how to assign binary values to the 03B1bets by dividing them into lower and upper halves. In practical implementations, we only store the assigned binary values; the symbols are shown here only to facilitate understanding of the wavelet tree decomposition process.

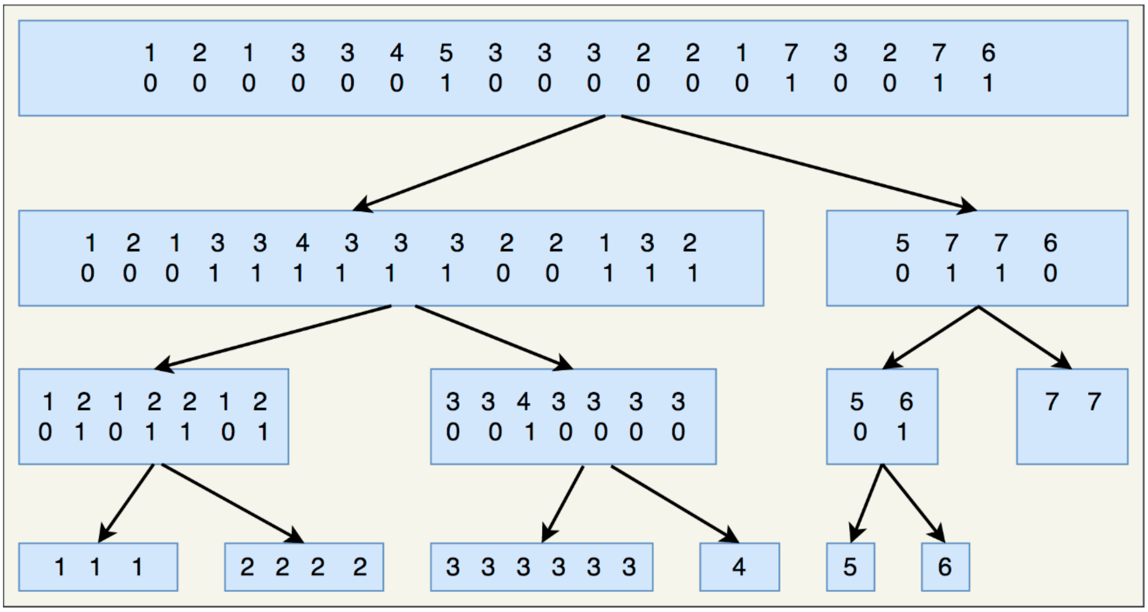


Figure 4. Wavelet tree example, with 0 for first half and 1 for second half of the 03B1bet.

To adaptively change the entropy measure during encoding, different variants can be used to obtain satisfactory compression and space results. The symbol assignment strategy can be changed to obtain better compression results. The wavelet tree is useful in many ways for data compression, especially with Huffman coding. By using a wavelet tree, we can obtain the desired bit pattern by using a specific encoding that change the tree shape.

Inserting or removing 03B1bet symbols in the original sequence requires changing the shape of the wavelet tree and the bitmaps that are stored at each level. This dynamism can be achieved when we combine the K^2-tree with the wavelet tree and it is very helpful for data security and compression.

4. Proposed Image Compression and Encryption Algorithm

Figure 5 illustrates the process of image compression and encryption by the proposed method. The whole process of compression and encryption was divided into three stages: At the first stage, compression and encryption were performed via lifting IWT and the SPIHT algorithm. In the second stage, compression of SPIHT output along with permutations and substitution operations was performed via a modified Kd-tree and a wavelet tree. Further compression was achieved through Huffman coding. Finally, at the third stage, encryption was performed on the output of the second stage by varying parameters in the logistic map and modified quadratic chaotic maps.

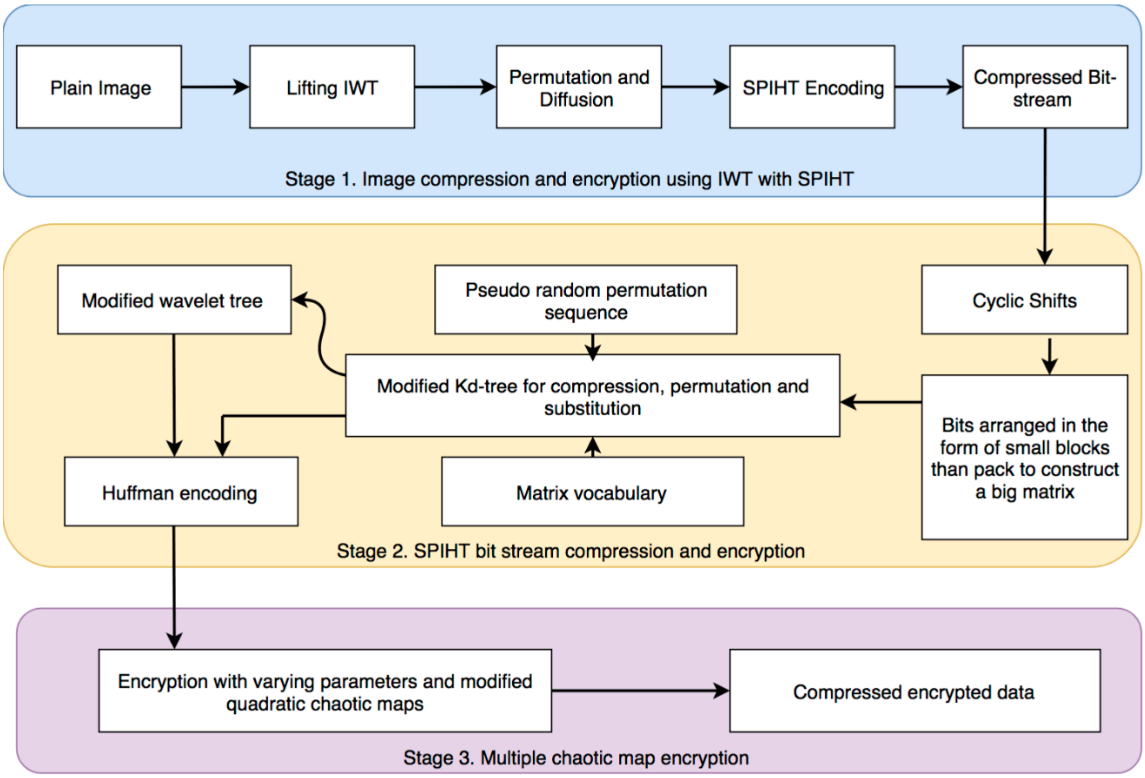

Plain Image
Lifting IWT
Permutation and Diffusion
SPIHT Encoding
Compressed Bit-stream
Stage 1. Image compression and encryption using IWT with SPIHT
Modified wavelet tree
Pseudo random permutation sequence
Cyclic Shifts
Modified Kd-tree for compression, permutation and substitution
Bits arranged in the form of small blocks than pack to construct a big matrix
Huffman encoding
Matrix vocabulary
Stage 2. SPIHT bit stream compression and encryption
Encryption with varying parameters and modified quadratic chaotic maps
Compressed encrypted data
Stage 3. Multiple chaotic map encryption

Figure 5. Joint image compression and encryption process.

4.1. Compression and Encryption Based on IWT and SPIHT

LWT with SPIHT yields the best results in terms of compression ratio and computational complexity with very good image quality for medical and natural images. The plain image is transformed into integer wavelet coefficients through LWT. The wavelet transform has the capability of placing a large percentage of the total signal energy in the low sub-band of the image. Therefore, the nonlinear inverse diffusion operations in Z_p on lower-band wavelet coefficients provide sufficient security and linear diffusion operations on the remaining coefficients to increase the 0 counts during SPIHT encoding. In this case, we only select the coefficient values in the 64 × 64 upper-left corner. Among them, we transformed two-byte positive $(0 - 65535)$ values through nonlinear inverse operations. However, unfortunately, 65536 is not a prime number. Hence, Z_{65536} is not a field and does not have a multiplicative inverse. Therefore, we choose the next prime, which is Z_{65537}, for taking the inverse. For the two bytes of the wavelet coefficient that is represented by m, the inverse operation can be defined as $I = m^{-1} mod(2^{16} + 1)$. After the diffusion process, these coefficients were permuted via a high-dimensional Kd-tree. These permuted and diffused coefficients were subjected to SPIHT for encoding.

4.2. Compression with Confusion-Diffusion through Kd-Tree, Wavelet Tree and Huffman Coding

In the second stage, we performed joint compression and encryption via Kd-tree, wavelet tree and Huffman coding. A detailed explanation is given below.

4.2.1. Confusion-Diffusion and Compression with K^2-Tree

Usually, SPIHT encoding produces a sparse bit stream. The sparseness can be increased by the linear diffusion of small wavelet coefficients with small negative value. Several cyclic shifts were performed on the SPIHT bit stream and reshaped into small blocks (2 × 2), packed to form large blocks (8 × 8) and combined into a large matrix (512 × 512). Then, this matrix was subjected to K^2-tree for further compression with confusion and diffusion operations. At different levels of K^2-tree, we can achieve permutations of sub-level matrices with the help of a pseudo-random permutation sequence (which is obtained via Matlab function randperm with rng(seed) with a specified seed value). In a modified K^2-tree, we achieved compression of homogeneous regions of zeros and ones at multiple levels. The matrix vocabulary was used to achieve diffusion at lower levels of the conceptual tree.

4.2.2. Diffusion and Compression with Wavelet Tree

Further compression along with confusion-diffusion was achieved via a modified wavelet tree by reordering L-array elements. We convert the L-array bits to symbols (2-bit symbols) and rearrange these symbols via the wavelet tree to achieve additional compression and diffusion.

4.2.3. Huffman Coding

Huffman encoding is the most widely used encoding technique and is the best known among the most popular techniques. In this technique, first, we calculate the frequency of each symbol with a probability distribution. Then, we assign code words based on the occurrence probability of each symbol. Huffman encoding is similar to a tree in which each node has either 0 or 2 children [41].

4.3. Chaotic-Map-Based Encryption

Logistic maps are one-dimensional maps, as expressed in Equation (6) and have been widely used in image encryption.

$$x_{n+1} = a x_n (1 - x_n) \qquad n = 0, 1, 2, \ldots N \tag{6}$$

where a is the parameter of the logistic map, and x_n is the sequence generated with Equation (6) with initial value x_0 $(0 < x_0 < 1)$. For 3.56999 < $a \leq 4$, Equation (6) becomes chaotic. From this chaotic function, real numbers can be obtained that form a pseudo-random sequence.

Research shows that logistic map sequences are not secure due to well-known weaknesses, such as small key size, short chaotic orbit and lack of resistance to phase-space reconstruction attacks. Therefore, to overcome these weaknesses, a logistic map with varying parameters (in Equation (7)) was proposed to expand the key space and improve resistance to phase-space construction attacks.

$$x_{k,n+1} = a_k x_{k,n} (1 - x_{k,n}) \qquad k = 1, 2, \ldots, M \qquad n = 0, 1, 2, \ldots N_k \tag{7}$$

Here, the value of k represents the specific parameter set and N_k represent the number of values in each set. $X_{k,n+1}$ is the output of the chaotic map at specific parameter value k. The a_k is the varying coefficients from [3.5699, 4] that can be vary either through other chaotic maps or with the periodic perturbation of the parameters like $a_k = 3.9 + 0.01 * k$. M is the cardinality of parameter set. The logistic chaotic map output with varying parameter sets can be represented as follows.

$X_{1,1}$; $X_{1,2}$;…..; $X_{1,N1}$	$X_{2,1}$; $X_{2,2}$;…..; $X_{2,N2}$	………..	$X_{i,1}$; $X_{i,2}$;…..; $X_{i,Ni}$	………..	$X_{M,1}$; $X_{M,2}$;….; $X_{M,Nk}$
k=1	k=2		k=i		k=M

The chaotic binary sequence was generated by the following Equation (8).

$$b_i = \begin{cases} 0 & x_i \leq 0.5 \\ 0 & x_i > 0.5 \end{cases} \qquad i = 0,\ 1, \ldots, n \tag{8}$$

The quadratic map is another fundamental type of chaotic system. The classical quadratic map can be represented as follows in Equation (9).

$$X_{n+1} = r - (X_n)^2 \tag{9}$$

Here, *r* is the chaotic parameter and *n* is the number of iterations. The quadratic map is a nonlinear chaotic system. It depends upon a mathematical equation; therefore, it is deterministic. However, it is highly sensitive to the initial conditions: a slight change in initial value x_0 can produce a significant change in the output of the map.

The Lyapunov exponent, which is denoted as λ, is a performance-measuring parameter of chaotic map systems. The value of the Lyapunov exponent indicates the sensitivity of the chaotic system to the initial condition quantitatively. For a discrete system and a chaotic orbit that starts with x_0, the Lyapunov exponent can be defined as follows in Equation (10):

$$\lambda(x_0) = \lim_{x \to \infty} \frac{1}{n} \ln\left|f'(x_i)\right| \tag{10}$$

Here, f' is the derivative of the function f'. From the above equation, we can calculate the value of λ, which can be positive, negative or zero. The value λ gives us important information about the state of the system. If the value of λ is negative, the system shows non-chaotic behavior; in contrast, if the value of λ is zero, then the system is in a steady state and cannot be used for a cryptosystem. Only if the value of λ is positive does the system show chaotic behavior and is sensitive to the initial conditions; however, the degree of sensitivity depends on the value of λ. The more positive the value of λ is, the more sensitive the system is to initial conditions and the closer to zero that λ is, the less sensitive the system will be to the initial conditions. Equations (11) and (12) are the modified quadratic maps that are proposed in.

$$X_{n+1} = \left(r + (1 - ax_n)^2\right)\text{mod}1 \tag{11}$$

$$X_{n+1} = \left(r + (1 - 8x_n)^2\right)\text{mod}1 \tag{12}$$

The authors replace $(X_n)^2$ in Equation (11) with the term $(1 - aX_n)^2$ and modulo 1, with *a* = 2, 4, 8.

A larger Lyapunov exponent indicates more-chaotic behavior. The above-modified quadratic chaotic map has Lyapunov exponent 3.4709, which is a very good value compared to other chaotic maps; thus, this map yields a chaotic sequence with desired properties.

4.4. Reverse Process of Compression and Encryption

The complete reverse process of compression and encryption is illustrated in Figure 6. This is the opposite process of the compression and encryption algorithm.

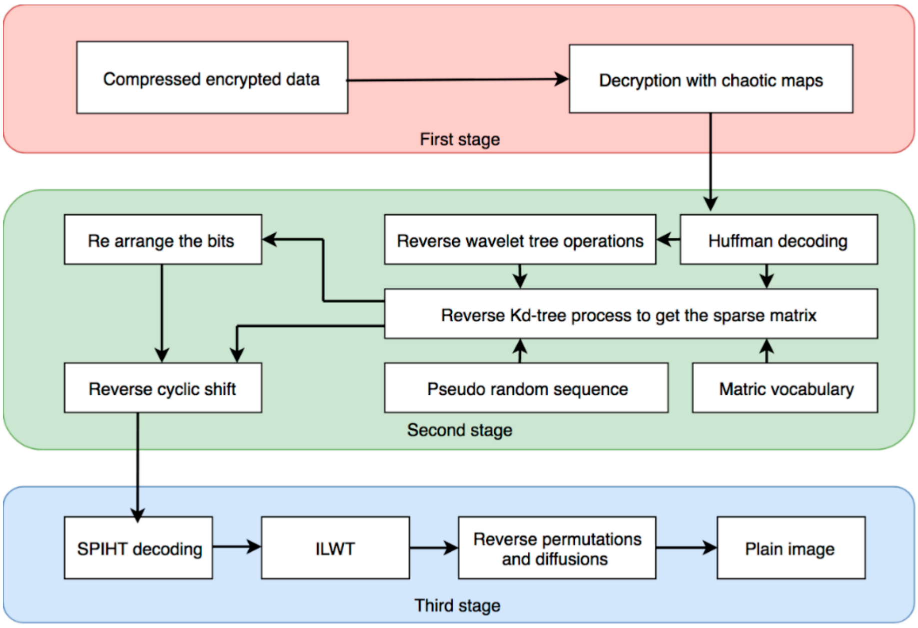

Compressed encrypted data
Decryption with chaotic maps
First stage
Re arrange the bits
Reverse wavelet tree operations
Huffman decoding
Reverse Kd-tree process to get the sparse matrix
Reverse cyclic shift
Pseudo random sequence
Matric vocabulary
Second stage
SPIHT decoding
ILWT
Reverse permutations and diffusions
Plain image
Third stage

Figure 6. Reverse encryption and compression process.

5. Experimental Results and Discussion

For the proposed hybrid compression and encryption algorithm, security tests and compression performance tests were performed. The experiments were implemented in the MATLAB R2016b programming environment developed by mathworks is an American privately held corporation, which was running on a personal computer with an Intel Corei3 3.5-GHz processor and 8-GB memory.

5.1. Experimental Results

The experiment was performed on six standard 512 × 512 grayscale images with 256 gray levels. The images are titled Baboon, Lena, Peppers, Airplane, House, and Lake. First, compression and encryption of these images were performed with IWT-based SPIHT. Then, this bit stream was further compressed along with maximum confusion diffusion operations via Kd-tree. Finally, an additional level of security was achieved via encryption through a varying parameter logistic map and modified quadratic chaotic maps. Experimental results demonstrate that our method provides maximum compression with a high level of security, which are the two core objectives in this research. Figure 7 shows Lena's encrypted results for the wavelet coefficients under various sizes of diffusion.

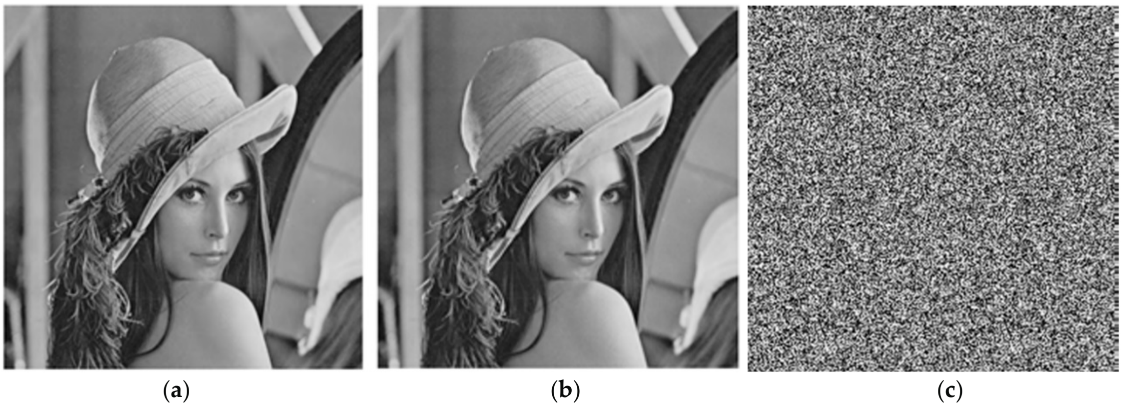


Figure 7. Experimental results. (**a**) Lena original grayscale image; (**b**) Lena reconstructed image; (**c**) Reconstructed with an incorrect key value.

5.2. Compression Performance

In this paper, lossless image compressions have been achieved. Table 1 lists the compression ratios (CRs) of six images when the permutation and diffusion operations were performed on the upper-left 64 × 64 wavelet coefficients and SPIHT bit-stream compression with permutations at block sizes of greater than 8×8. The compression performance is represented by the compression ratio (CR) or bits per pixel (bpp). Larger values of CR indicate more compression, whereas lower values of bpp indicate more compression. CR and bpp are defined as follows.

$$\mathrm{CR} = \frac{\mathrm{Number} - \mathrm{of} - \mathrm{bits} - \mathrm{of} - \mathrm{plain} - \mathrm{image}}{\mathrm{Number} - \mathrm{of} - \mathrm{bits} - \mathrm{of} - \mathrm{cipher} - \mathrm{image}}$$

$$\mathrm{bpp} = \frac{\mathrm{Total} - \mathrm{bits} - \mathrm{in} - \mathrm{compression} - \mathrm{image}}{\mathrm{Total} - \mathrm{number} - \mathrm{of} - \mathrm{image} - \mathrm{pixels}}$$

Here, we used both the CR and bpp for comparison.

The compression comparison of the proposed method with the most relevant methods [14,35,37,38] indicate that our method has achieved much more compression performance in joint image compression and encryption domain as compared to the existing methods. It indicates that the algorithm has a better compression ratio and the impact of encryption on compression is not substantial; thus, the proposed approach achieves a good balance between compression and encryption.

The commonly used criteria for image quality assessment after compression are mean square error (MSE), which is defined in Equation (13), and peak signal-to-noise ratio (PSNR), which is defined in Equation (14).

$$MSE = \frac{1}{M \times N} \sum_{j=1}^{N} \left(\overset{\frown}{I}(i,j) - I(i,j) \right) \tag{13}$$

Here, $\overset{\frown}{I}(i,j)$ denotes the original image and $I(i,j)$ is the testing image, i.e., the compressed image. M and N represent the height and width, respectively, of the image.

$$PSNR = 10 \times \log_{10} \left(\frac{x_{peak}^{2}}{MSE} \right) \tag{14}$$

Here, x_{peak} represents the peak signal value.

Table 1. CR and PSNR values of our approach and the compared approaches.

Image	**PSNR/dB**			**CR/bpp**				
	Our Approach	**Ref. [37]**	**Ref. [38]**	**Our Approach**	**Ref. [14] CR**	**Ref. [35] CR**	**Ref. [37] CR**	**Ref. [38] QF = 80/bpp**
Lena (grayscale)	39.85	42.23	—	8.19/0.97	6.222713	10.426	4	-
Baboon (grayscale)	36.37	35.15	32.5956	8.20/0.97	7.569092	4.977	4	2.4414
Peppers (grayscale)	37.81	41.3	-	8.26/0.96	6.698654	-	4	-
Airplane (grayscale)	40.68	-	-	8.49/0.94	6.388039	5.313	-	-
House (grayscale)	40.41	-	-	8.13/0.98	6.671848	-	-	-
Lake (grayscale)	34.54	-	-	8.39	-	-	-	-

The results in Table 1 demonstrate that the proposed method yields very good compression results compared to other related joint image compression and encryption methods with high-quality images.

5.3. Key-Space Analysis

Sufficiently large key size is necessary for any good encryption algorithm. In the proposed method, we have used four keys: wavelet coefficient permutation and diffusion keys, SPIHT bit-stream cyclic shift with permutation and matrix vocabulary keys, logistic chaotic map keys and quadratic chaotic map keys. At each stage, we have used 4-real-number keys of double data type. In these four stages, keys that are composed of a minimum of 16 real numbers are used. These 16 double real numbers provide a key size of 1024 bits. Thus, the size of the key space is 2^{1024}, which is large enough to resist any brute-force attack. Table 2 shows the comparison among key sizes of the proposed and pseudorandom number based encryption schemes.

Table 2. Key size comparison with some novel pseudorandom number generators.

Proposed Method	Ref. [8]	Ref. [9]	Ref. [10]	Ref. [11]
2^{1024}	2^{183}	2^{776}	2^{320}	2^{96}

5.4. Key Sensitivity

Key sensitivity analysis determine the affect of small change in the encryption key on the ciphered data. The encryption scheme in which small change in the encryption key leads to considerable difference in cipher text considered to be a good encryption scheme. The proposed joint image compression and encryption scheme used at least four keys. For the diffusion and permutations of the lower band 64 × 64 integer wavelet coefficients. A tiny change in the permutation key affects the whole SPIHT encoding. Similarly a one bit change in the permutations sequence at any level of Kd-tree applied on the SPIHT output changed completely the bit stream. The other two keys are varying parameters logistic and quadratic chaotic maps keys which are also very sensitive to the change. Here, the key sensitivity results are presented in the form of PSNR and mean structural similarity index (MSSIM) after apply tiny changes at different levels of keys. The PSNR of the wrong key deciphered image was calculated thorugh Equation (14), SSIM and MSSIM were obtained with Equations (15) and (16) respectively. Here μ and σ are the mean and variance respectively. Table 3 shows the key sensitivity test results of the proposed scheme.

$$SSIM(x,y) = \frac{(2m_x m_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{15}$$

$$MSSIM(X,Y) = \frac{1}{n}\sum_{i=1}^{n} SSIM(x_i, y_i) \tag{16}$$

Table 3. PSNR and MSSIM of the decrypted image with wrong key.

Image		**Decryption with 1-bit Key Difference in the Permutations of Lower 64 × 64 Wavelet Coefficients**	**Decryption with 1-bit Key Difference in the Permutations of SPIHT Output with Kd-Tree**	**Decryption with 1-bit Key Difference in Logistic Chaotic Map**	**Decryption with 1-bit Key Difference in Quadratic Chaotic Map**
Lena	PSNR	9.5241	3.0214	2.0135	2.8742
	MSSIM	0.1472	0.0128	0.0214	0.1251
Baboon	PSNR	7.3254	4.5210	2.0176	2.6522
	MSSIM	0.0713	0.0843	0.0421	0.2414
Pepper	PSNR	7.3617	3.2412	2.0187	2.6415
	MSSIM	0.1382	0.0421	0.0342	0.1453
Airplane	PSNR	8.3692	3.0124	2.0873	2.3541
	MSSIM	0.1426	0.0621	0.0462	0.3524
House	PSNR	7.3641	2.0146	3.5214	2.6521
	MSSIM	0.1536	0.0143	0.0517	0.3245
Lake	PSNR	0.7125	3.0241	2.0364	2.6521
	MSSIM	0.1285	0.0142	0.0394	0.0124

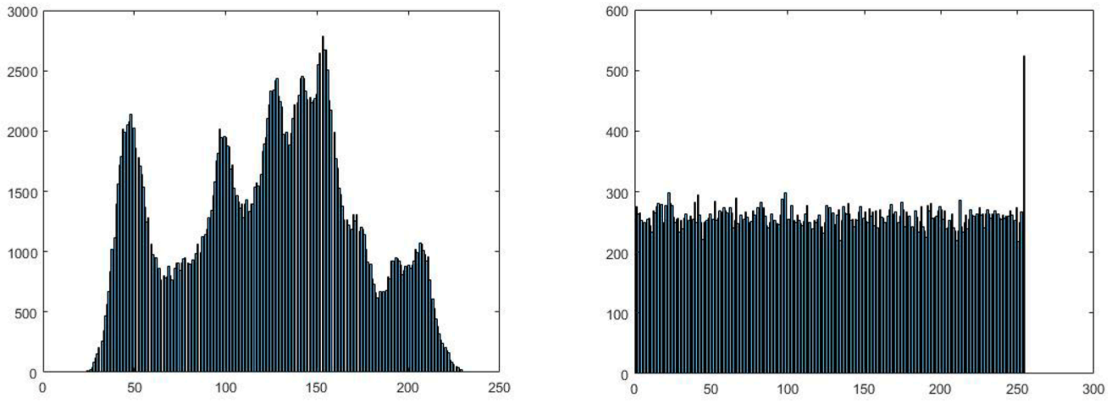
5.5. Resistance to Statistical Attacks

Resistance to statistical attacks is the most basic objective of an image encryption algorithm. Shannon suggested that maximum confusion and diffusion should be employed in a cryptosystem to remove the correlation characteristics of image pixels to frustrate powerful statistical analysis. In the proposed compression and encryption algorithm, we achieve maximum confusion and diffusion at various levels via nonlinear inverse operations, K^2-tree and multiple chaotic maps in the last stage. After many confusion and diffusion operations were performed through the joint compression and encryption operations, the cipher image shows a truly random distribution. This random distribution of the cipher image was evaluated using histograms, adjacent-pixel correlations, and the information entropy of the cipher image.

5.5.1. Histogram Analysis

Figure 8 shows the "Lena" image histograms and, on the right, the corresponding "Lena" cipher image. Figure 8 shows that the histogram of the corresponding cipher image is almost uniformly distributed; therefore, this technique can withstand all types of statistical attacks. We also analyze the distribution of the pixel values of encrypted images through the chi-square test. The chi-square value was calculate using Equation (17).

$$\chi^2 = \sum_{i=0}^{255} \frac{(O_i - E)^2}{E} \tag{17}$$



(**a**) (**b**)

Figure 8. Histogram analysis. (**a**) Original Lena image histogram; (**b**) Encrypted Lena image histogram.

Here, O_i is the observed frequency of the ith pixel i $(0 \leq i \leq 255)$, E is the expected frequency of pixel value i, and $v_0 = (m \times n)/256$. The results of the χ^2-test for the 6 pairs of plain/encrypted test images are presented in Table 4:

Table 4. Chi-squared test results for histogram uniformity.

Image	χ^2-Text	
	Plain Image	**Encrypted Image**
Lena (grayscale)	158360	270
Baboon (grayscale)	211370	281
Peppers (grayscale)	549150	276
Airplane (grayscale)	728690	286
House (grayscale)	880954	291
Lake (grayscale)	181220	273

From Table 4, for the six images, the values of the χ^2-test are all lower than the critical value. Thus, we conclude that the distribution of pixel values in the encrypted image is uniform. That means that the proposed scheme can resist statistical attacks.

5.5.2. Correlation of Two Adjacent Pixels

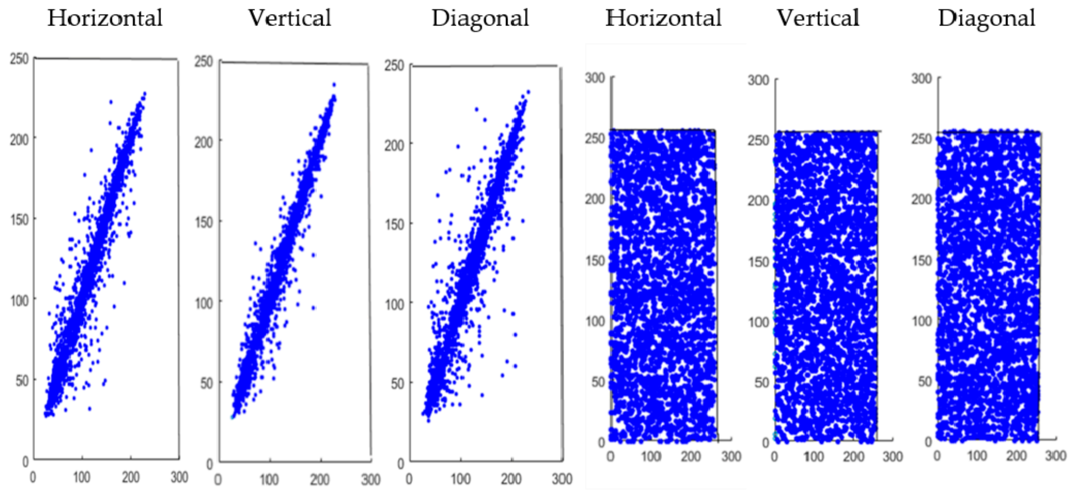
In image data, adjacent pixels show very strong correlations with each other. Good encryption algorithms must remove this strong correlation to resist statistical attacks. The correlation among adjacent pixels calculated via Equation (18).

$$C_r = \frac{N\sum_{j=1}^{N}(x_j y_j) - \left(\sum_{j=1}^{N} x_j\right)\left(\sum_{j=1}^{N} y_j\right)}{\left(N\sum_{j=1}^{N} x_j^2 - \left(\sum_{j=1}^{N} x_j\right)^2\right)\left(N\sum_{j=1}^{N} y_j^2 - \left(\sum_{j=1}^{N} y_j\right)^2\right)} \quad (18)$$

The results that were obtained from the above relation demonstrate that the proposed algorithm efficiently destroys the correlation among adjacent pixels, as shown in Table 5 and Figure 9. Therefore, no information can be obtained through the correlation of the pixels.

Table 5. Correlation coefficients of adjacent pixels.

Image	**Orientation**		
	Horizontal	**Vertical**	**Diagonal**
Lena (plain)	0.9719	0.9850	0.9593
Lena (encrypted)	0.00051	−0.0043	0.00031
Baboon (plain)	0.9337	0.9123	0.8669
Baboon (encrypted)	−0.00245	−0.00314	0.00137
Peppers (plain)	0.9802	0.9803	0.9706
Peppers (encrypted)	−0.00178	0.00812	−0.00031
Airplane (plain)	0.9676	0.9628	0.9371
Airplane (encrypted)	0.00519	−0.00362	−0.00039
House (plain)	0.9401	0.9372	0.8952
House (encrypted)	−0.00238	−0.00497	0.00847
Lake (plain)	0.9761	0.9764	0.9616
Lake (encrypted)	0.00874	0.00812	0.00038



(**a**)

(**b**)

Figure 9. (**a**). Distribution of Lena image; (**b**). Distribution of encrypted Lena image.

5.5.3. Information Entropy Analysis

The information entropy is a measure of the data distribution after encryption, which is defined by Equation (19). If each symbol has almost the same entropy as a random distribution, then the system has high random complexity and each pixel is treated as a random event.

$$H(S) = \sum_{S} P(S_i) \log_2 \frac{1}{P(S_i)} \tag{19}$$

Here, $P(S_i)$ represents the probability of each symbol S_i. According to Equation (19), for a uniform distribution, the ideal entropy for each symbol is 8 for an image in which each pixel consists of 8 bits. Table 6 shows the entropy analysis results for encrypted images. Table 6 shows global entropies of the encrypted images. The results indicates that the entropy values are very close to the ideal value 8 for grayscale images.

Table 6. Global information entropy of encrypted images.

Image (Encrypted)	Information Entropy
Lena	7.99642
Baboon	7.99645
Pepper	7.98953
Airplane	7.99231
House	7.99483
Lake	7.99681

Encrypted image randomness measured with traditional methods only provide quantitative measure rather than qualitative extent. Ref. [42] developed a new test to measure the randomness of an encrypted image over local blocks that is known as local Shannon entropy test. The local Shannon entropy is defined as follows.

$$\overline{H_{(K,T_B)}}(S) = \sum_{i=1}^{k} \frac{H(S_i)}{k} \tag{20}$$

Here, $\overline{H_{(K,T_B)}}(S)$ represents the local entropy with K blocks of TB number of pixels in each block and S1, S2, ..., Sk are randomly chosen non-overlapping encrypted image blocks. We have selected 30 blocks (k = 30) from six test images randomly with T_B = 1936 and calculate local entropy test results, which are shown in Table 7.

Table 7. Local information entropy of encrypted images (k = 30, T_B = 1936, α = 0.05).

Image (Encrypted)	Information Entropy	Result
Lena	7.90642	Pass
Baboon	7.90382	Pass
Pepper	7.91124	Pass
Airplane	7.90235	Pass
House	7.90321	Pass
Lake	7.91156	Pass
Mean ± Std.	7.90643 ± 0.0040	

5.5.4. NIST SP800-22 Tests of Encrypted Bit Stream

NIST tests SP800-22 are used to detect the randomness of the sequence [43]. This test works on binary data by converting it in the form of ±1 and calculate the absolute sum. The test statistic is obtained as follows by Equation (21).

$$S_{obs} = \frac{|S_n|}{\sqrt{n}} \tag{21}$$

The value of P determine the randomness of a sequence. If the value of P is less than 0.01 the sequence is considered to be non-random otherwise random. The test results on the encrypted bit stream are shown in Table 8.

$$P - value = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right) \tag{22}$$

Table 8. NIST SP800-22 tests of encrypted bit stream.

Test	p-Value	Pass Rate
Monobit test	0.231475	Pass
Frequency within block test	0.754126	Pass
Runs test	0.632514	Pass
Longest run ones in a block test	0.641258	Pass
Binary matrix rank test	0.412572	Pass
DFT test	0.754163	Pass
Non overlapping template matching test	0.425712	Pass
Overlapping template matching test	0.912458	Pass
Universal test	0.952413	Pass
Linear complexity test	0.342587	Pass
Serial test	0.765812	Pass
Approximate entropy test	0.672541	Pass
Cumulative sum test	0.192745	Pass
Random excursion test	0.256312	Pass
Random excursion variant test	0.028735	Pass

5.6. Resistance to Linear and Differential Attacks

The proposed compression and encryption scheme have multiple levels of securities in series. First, integer wavelet coefficients permutations and substitutions were performed by which the SPIHT encoding generates completely different bit stream, after that the produced bit stream under goes extensive permutations and substitutions with Kd-tree. Finally, multiple chaotic maps-based encryptions achieved randomness in the bit stream. The proposed joint image compression and encryption algorithm destroy the relationship between the original image and the encrypted image. The experimental results demonstrate that the scheme is sufficiently secure against all known types of attacks, such as brute-force attack, differential attack, chosen plaintext attack, known-plaintext attack and statistical attack. Resistant to differential attacks can be checked through following two security tests. The values of number of pixels change rate (NPCR) and unified average change intensity (UACI) values used to evaluate these test results. The NPCR is defined as follows in Equation (23).

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{23}$$

Here, $D(i,j)$ is the difference value of the two corresponding pixels of two images and $W \times H$ size of the image. The value of NPCR gets closer to 100% then it became more sensitive to the change in the plain image. The comparison between the corresponding pixels of the two encrypted images is done as follows

$$D(i,j) = \begin{cases} 1 & C_1(i,j) \neq C_2(i,j) \\ 0 & C_1(i,j) = C_2(i,j) \end{cases} \tag{24}$$

Here, C_1 is the original encrypted image and C_2 is the encrypted image after changing one pixel value in the original image.

The UACI measure the change between original image and the corresponding encrypted image. Large values of UACI strongly resist to the differential attacks. UACI is defined as follow by Equation (25).

$$UACI = \frac{1}{W \times H}\left[\frac{C_1(i,j) - C_2(i,j)}{255}\right] \times 100\% \tag{25}$$

Table 9 shows the results of NPCR and UACI tests. For good encryption algorithms that can resist differential attacks the values of NPCR and UACI tests are around 99.6094% and 33.4635% respectively [44].

Table 9. NPCR and UACI results.

Images/Tests	Lena	Baboon	Pepper	Airplane	House	Lake
NPCR	99.47%	99.36%	99.52%	99.38%	99.25%	99.46%
UACI	33.23%	33.21%	33.34%	33.46%	33.14%	33.18%

5.7. Computational Compexity and Encryption Speed

The computational complexity of the proposed scheme for an image with n number of pixels is depicted in the Table 10. The lifting integer wavelet transform reduces the computational time by almost 50% compared to the traditional wavelet transform and the SPIHT encoding algorithm is the most computationally efficient compression algorithm. Only Kd-tree takes more time when applied on the whole data set, compared to other compression and encryption schemes. However, many researchers have proposed fully parallel Kd-tree implementations in other big-data applications; hence, the processing time can be reduced in the future by implementing a fully parallel Kd-tree for this scheme. The chaotic map encryption is computationally very light weight as like XOR operations mostly. Table 11 lists the compression and encryption speeds of the proposed scheme.

Table 10. Computational complexity.

Joint Compression and Encryption	Diffusion and Permutations of Wavelet Coefficients	SPIHT Encoding and Bits Packing	Permutations with Kd-Tree and Chaotic Bit Stream Encryption
Computational complexity	$O(n)$	$O(n)$	$O(n \times \log(n))$

Table 11. Compression and Encryption Times.

Image	Wavelet Coefficient Encryption Time	SPIHT Compression Time	Compression and Encryption with Kd-Tree	Compression with Wavelet Tree and Huffman Coding	Total Compression and Encryption Time
Lena	3.3 s	2.4 s	12.3 s	1.2 s	19.2 s
Baboon	3.1 s	2.1 s	12.1 s	1.4 s	18.7 s
Peppers	3.4 s	2.3 s	12.5 s	1.3 s	19.5 s
Airplane	3.0 s	2.1 s	12.3 s	1.2 s	18.6 s
House	3.1 s	2.2 s	12.4 s	1.4 s	19.1 s
Lake	3.2 s	2.3 s	12.2 s	1.3 s	19.0 s

5.8. Joint Compression and Encryption for the Worst Case Images

The proposed scheme was evaluated with complete white and black images. As the complete white and black images have no information and in the trnsform domain appear only as a dot. Therefore, wavelet coefficients only have few information in the form of only four values. However, if these values passed to SPIHT encoding, it generates almost the same number of ones and zeros as for the other iamges. The rest of the procedure have no difference for worst case and other images.

5.9. Performance Comparison with Related Methods

We have compared our proposed method with most of the related joint compression and encryption methods. The experimental results demonstrate that our method outperforms the existing methods in terms of compression, as shown in Table 1, and security, according to the statistical results. Joint compression and encryption of a SPIHT bit stream with modified Kd-tree and varying parameters chaotic maps provide efficient security with compressed data.

6. Conclusions

In this paper, a joint image compression and encryption algorithm that is based on IWT with SPIHT, Kd-tree and multiple chaotic maps was proposed. First, compression is performed via lifting-based IWT and SPIHT. The diffusion of the major coefficients of IWT is carried out through nonlinear inverse operations. Permutations of the lower-frequency-band coefficients are performed via Kd-tree. Then, this SPIHT bit stream is subjected to Kd-tree for further compression, permutations and diffusions. In Kd-tree, permutations are performed with pseudo-random sequences. The diffusion of leaf nodes is achieved through matrix vocabulary. The Kd-tree output is further compressed via a modified wavelet tree and Huffman coding. Using this method, we achieved a high compression ratio and maximum confusion and diffusion operations. The main advantage of this scheme is that, with a robust security level, the compression performance is not degraded, as is the case with most joint compression and encryptions schemes. Finally, this bit stream is encrypted via varying parameter logistic maps and modified quadratic maps. Multi-level security is achieved that can withstand cryptanalysis and all types of known attacks. Moreover, the key space is very large, which makes the scheme resistant to brute-force attacks. The test results demonstrate that the system is sufficiently secure with satisfactory lossless compression performance.

Author Contributions: Conceptualization and experiments, N. and J.S.; Data curation, J.S. and M.A.A.; Data analysis, N. and B.C.; Writing-Original Draft Preparation, N. and J.S.; Writing-Edit and Review, H.X., J.S. and H.H.

Acknowledgments: This work was supported by National Key R &D Program of China (No. 2017YFB0802400) and Chongqing Research Program of Basic Science and Frontier Technology (No. cstc2017jcyjB0305).

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

References

1. Yuen, C.-H.; Wong, K.-W. A chaos-based joint image compression and encryption scheme using DCT and SHA-1. *Appl. Soft Comput.* **2011**, *11*, 5092–5098. [CrossRef]
2. Wang, B.; Zheng, X.; Zhou, S.; Xhou, C.; Wei, X.; Zhang, Q.; Che, C. Encrypting the compressed image by chaotic map and arithmetic coding. *Opt. Int. J. Light Electron Opt.* **2014**, *125*, 6117–6122. [CrossRef]
3. Lian, S. Efficient image or video encryption based on spatiotemporal chaos system. *Chaos Solitons Fractals* **2009**, *40*, 2509–2519. [CrossRef]
4. Li, S.; Mou, X.; Cai, Y.; Ji, Z.; Zhang, J. On the security of a chaotic encryption scheme: Problems with computerized chaos in finite computing precision. *Comput. Phys. Commun.* **2003**, *153*, 52–58. [CrossRef]
5. Gong, L.; Deng, C.; Pan, S.; Zhou, N. Image compression-encryption algorithms by combining hyper-chaotic system with discrete fractional random transform. *Opt. Laser Technol.* **2018**, *103*, 48–58. [CrossRef]
6. Singh, R.K.; Kumar, B.; Shaw, D.K.; Khan, D.A. Level by level image compression-encryption algorithm based on quantum chaos map. *J. King Saud Univ. Comput. Inf. Sci.* 2018. [CrossRef]
7. Wang, Q.; Wei, M.; Chen, X.; Miao, Z. Joint encryption and compression of 3D images based on tensor compressive sensing with non-autonomous 3D chaotic system. *Multimed. Tools Appl.* **2018**, *77*, 1715–1734. [CrossRef]
8. Stoyanov, B.; Kordov, K. Novel secure pseudo-random number generation scheme based on two tinkerbell maps. *Adv. Stud. Theor. Phys.* **2015**, *9*, 411–421. [CrossRef]
9. Tong, X.; Liu, Y.; Zhang, M.; Xu, H.; Wang, Z. An image encryption scheme based on hyperchaotic Rabinovich and exponential chaos maps. *Entropy* **2015**, *17*, 181–196. [CrossRef]
10. Stoyanov, B.; Kordov, K. A novel pseudorandom bit generator based on Chirikov standard map filtered with shrinking rule. *Math. Prob. Eng.* **2014**, *2014*, 986174. [CrossRef]
11. Stoyanov, B.P. Chaotic cryptographic scheme and its randomness evaluation. *AIP Conf. Proc.* **2012**, *1487*, 397–404.
12. Liu, Y.; Tong, X. Hyperchaotic system-based pseudorandom number generator. *IET Inf. Secur.* **2016**, *10*, 433–441. [CrossRef]

13. Wu, Y.; Noonan, J.P.; Yang, G.; Jin, H. Image encryption using the two-dimensional logistic chaotic map. *J. Electron. Imaging* **2012**, *21*, 013014. [CrossRef]
14. Zhang, M.; Tong, X. Joint image encryption and compression scheme based on IWT and SPIHT. *Opt. Lasers Eng.* **2017**, *90*, 254–274. [CrossRef]
15. Hamdi, M.; Rhouma, R.; Belghith, S. A selective compression-encryption of images based on SPIHT coding and Chirikov Standard Map. *Signal Proc.* **2017**, *131*, 514–526. [CrossRef]
16. Xiang, T.; Qu, J.; Xiao, D. Joint SPIHT compression and selective encryption. *Appl. Soft Comput.* **2014**, *21*, 159–170. [CrossRef]
17. Said, A.; Pearlman, W.A. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.* **1996**, *6*, 243–250. [CrossRef]
18. Daubechies, I.; Sweldens, W. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.* **1998**, *4*, 247–269. [CrossRef]
19. Rhouma, R.; Belghith, S. Cryptanalysis of a spatiotemporal chaotic image/video cryptosystem. *Phys. Lett. A* **2008**, *372*, 5790–5794. [CrossRef]
20. Rhouma, R.; Belghith, S. Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Phys. Lett. A* **2008**, *372*, 5973–5978. [CrossRef]
21. Alvarez, G.; Amigó, J.M.; Arroyo, D.; Li, S. Lessons learnt from the cryptanalysis of chaos-based ciphers. In *Chaos-Based Cryptography: Theory, Algorithms and Applications*; Springer: Cham, Switzerland, 2011; Volume 354, pp. 257–295.
22. Alvarez, G.; Li, S. Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 3743–3749. [CrossRef]
23. Li, C.; Li, S.; Lo, K.-T. Breaking a modified substitution–diffusion image cipher based on chaotic standard and logistic maps. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 837–843. [CrossRef]
24. Li, C.; Xie, T.; Liu, Q.; Cheng, G. Cryptanalyzing image encryption using chaotic logistic map. *Nonlinear Dyn.* **2014**, *78*, 1545–1551. [CrossRef]
25. Memon, Q.A. Synchronized choas for network security. *Comput. Commun.* **2003**, *26*, 498–505. [CrossRef]
26. Ravichandran, D.; Praveenkumar, P.; Rayappan, J.B.B.; Amirtharajan, R. Chaos based crossover and mutation for securing DICOM image. *Comput. Biol. Med.* **2016**, *72*, 170–184. [CrossRef] [PubMed]
27. El-Latif, A.A.A.; Li, L.; Zhang, T.; Wang, N.; Song, X.; Niu, X. Digital image encryption scheme based on multiple chaotic systems. *Sens. Imaging Int. J.* **2012**, *13*, 67–88. [CrossRef]
28. Alvarez, G.; Li, S.; Hernandez, L. Analysis of security problems in a medical image encryption system. *Comput. Biol. Med.* **2007**, *37*, 424–427. [CrossRef] [PubMed]
29. Ravichandran, D.; Praveenkumar, P.; Rayappan, J.B.B.; Amirtharajan, R. DNA chaos blend to secure medical privacy. *IEEE Trans. Nanobiosci.* **2017**, *16*, 850–858. [CrossRef] [PubMed]
30. Zhang, X.; Nie, W.; MA, Y.; Tian, Q. Cryptanalysis and improvement of an image encryption algorithm based on hyper-chaotic system and dynamic S-box. *Multimed. Tools Appl.* **2017**, *76*, 15641–15659. [CrossRef]
31. Liu, L.; Miao, S. An image encryption algorithm based on Baker map with varying parameter. *Multimed. Tools Appl.* **2017**, *76*, 16511–16527. [CrossRef]
32. Liu, L.; Miao, S. A new image encryption algorithm based on logistic chaotic map with varying parameter. *SpringerPlus* **2016**, *5*, 289. [CrossRef] [PubMed]
33. Maqbool, S.; Ahmad, N.; Muhammad, A.; Enriquez, A.M.M. Simultaneous Encryption and Compression of Digital Images Based on Secure-JPEG Encoding. In *Mexican Conference on Pattern Recognition*; Springer: Cham, Switzerland, 2016; pp. 145–154.
34. Ji, X.; Bai, S.; Zhu, G.; Yan, B. Image encryption and compression based on the generalized knight's tour, discrete cosine transform and chaotic maps. *Multimed. Tools Appl.* **2017**, *76*, 12965–12979.
35. Tong, X.-J.; Chen, P.; Zhang, M. A joint image lossless compression and encryption method based on chaotic map. *Multimed. Tools Appl.* **2017**, *76*, 13995–14020. [CrossRef]
36. Zhang, X.; Wang, X. Chaos-based partial encryption of SPIHT coded color images. *Signal Proc.* **2013**, *93*, 2422–2431. [CrossRef]
37. Zhang, M.; Tong, X. A new algorithm of image compression and encryption based on spatiotemporal cross chaotic system. *Multimed. Tools Appl.* **2015**, *74*, 11255–11279. [CrossRef]
38. Li, P.; Lo, K.-T.A. Content-Adaptive Joint Image Compression and Encryption Scheme. *IEEE Trans. Multimed.* **2018**, *20*, 1960–1972. [CrossRef]

39. De Bernardo, G.; Álvarez-García, S.; Brisaboa, N.R.; Navarro, G.; Pedreira, O. Compact querieable representations of raster data. In *International Symposium on String Processing and Information Retrieval*; Springer: Cham, Switzerland, 2013; pp. 96–108.

40. Navarro, G. Wavelet trees for all. In *Annual Symposium on Combinatorial Pattern Matching*; Springer: Cham, Switzerland, 2012; pp. 2–26.

41. Huffman, D.A. A method for the construction of minimum-redundancy codes. *Proc. IRE* **1952**, *40*, 1098–1101. [CrossRef]

42. Wu, Y.; Zhou, Y.; Saveriades, G.; Agaian, S.; Noonan, J.P.; Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inf. Sci.* **2013**, *222*, 323–342. [CrossRef]

43. Bassham, L.E.; Rukhin, A.L.; Soto, J.; Nechvatal, J.R.; Smid, M.E.; Barker, E.B.; Leigh, S.D.; Levenson, M.; Vangel, M.; Banks, D.L.; et al. *Sp 800-822 rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; National Institute of Standards & Technology: Gaithersburg, MD, USA, 2010.

44. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI randomness tests for image encryption. Cyber journals: Multidisciplinary journals in science and technology. *J. Sel. Areas Telecommun.* **2011**, *1*, 31–38.