Hindawi
Security and Communication Networks
Volume 2017, Article ID 7367518, 13 pages
https://doi.org/10.1155/2017/7367518

Research Article

Two-Phase Image Encryption Scheme Based on FFCT and Fractals

Mervat Mikhail, Yasmine Abouelseoud, and Galal ElKobrosy

Department of Engineering Mathematics, Faculty of Engineering, Alexandria 21544, Egypt

Correspondence should be addressed to Mervat Mikhail; mervat.mikhail80@gmail.com

Received 19 July 2016; Revised 21 October 2016; Accepted 20 November 2016; Published 22 January 2017

Academic Editor: Anna Cinzia Squicciarini

This paper blends the ideas from recent researches into a simple, yet efficient image encryption scheme for colored images. It is based on the finite field cosine transform (FFCT) and symmetric-key cryptography. The FFCT is used to scramble the image yielding an image with a uniform histogram. The FFCT has been chosen as it works with integers modulo p and hence avoids numerical inaccuracies inherent to other transforms. Fractals are used as a source of randomness to generate a one-time-pad keystream to be employed in enciphering step. The fractal images are scanned in zigzag manner to ensure decorrelation of adjacent pixels values in order to guarantee a strong key. The performance of the proposed algorithm is evaluated using standard statistical analysis techniques. Moreover, sensitivity analysis techniques such as resistance to differential attacks measures, mean square error, and one bit change in system key have been investigated. Furthermore, security of the proposed scheme against classical cryptographic attacks has been analyzed. The obtained results show great potential of the proposed scheme and competitiveness with other schemes in literature. Additionally, the algorithm lends itself to parallel processing adding to its computational efficiency.

1. Introduction

Multimedia communication has now become a common practice in our daily lives. Multimedia data demand both high transmission rates and security. Medical imaging systems, military image databases, and pay-per-view TV are examples of applications in which security is an essential requirement of the multimedia system [1].

Encryption is a key component for secure communication. It is the cryptographic primitive responsible for converting data into a form that is only intelligible by the intended recipient. A secret piece of information held by the recipient, the decryption key, is used to recover the original message. The proper choice of such a key is a critical component to any cryptosystem, especially in stream ciphers [2].

Image encryption is far more challenging compared to textual data encryption. The adjacent bits in an image usually show high correlation which can be used in cryptanalysis. Moreover, the image size is a major challenge requiring highly efficient algorithms that can handle such huge data size in an acceptable time frame. Furthermore, traditional symmetric encryption schemes do not work well for encrypting images. Several attempts have been made to adapt standard encryption schemes to suit the delicate nature of images [3]. The enciphered image should look random with a uniform histogram and passing well-known statistical tests of randomness in the NIST suite [4]. Moreover, the encryption scheme should be sensitive to small variations in the plain image showing a significant change in the enciphered image.

Image encryption has thus attracted the attention of many researchers, who attempt to develop new schemes representing different tradeoffs between the complexity of the algorithm and its reliability. In this paper, an image encryption algorithm is proposed employing both the finite field cosine transform [5] and fractal images [6]. Fractal images can be easily generated and they are used as a source of randomness for constructing a strong stream cipher key. The preliminary results show that the proposed scheme shows comparable performance to other schemes in literature.

The rest of the paper is organized as follows. In the following section, related work on image encryption in literature is reviewed. The finite field cosine transform, zigzag scanning idea, and fractal images are defined in Section 3. Some important characteristics of the FFCT are briefly discussed and the transform matrix to be used in our algorithm is presented. In

Section 4, evaluation techniques used to test the developed algorithm are described. Our image encryption scheme is introduced in Section 5. System key, complexity, and speed analysis are provided in Sections 6, 7, and 8, respectively. The experimental results used to illustrate and evaluate the performance of our approach are presented in Section 9. Resistance of the proposed encryption scheme to classical cryptographic attacks, like known-plaintext and chosen-plaintext attacks, is examined in Section 10. A comparative study between the proposed scheme and other schemes in literature is presented in Section 11. Finally, Section 12 concludes the paper.

2. Related Work

Much attention has been lately devoted to developing efficient and highly secure image encryption schemes. Image encryption algorithms are classified into three major categories [7]: (i) pixel position permutation based algorithms, in which a pixel is replaced by another pixel of the same image, (ii) value transformation based algorithms, in which a pixel is converted to another pixel value, and (iii) visual transformation based algorithms, in which another image is superimposed on another image such as using an image as a key or watermark-based encryption. Now, hybrid algorithms are most dominating. Different types of permutation or shuffling techniques have been successfully applied. Moreover, numerous frequency domain transforms have been considered. Furthermore, there is abundant literature on the use of chaotic maps as well as the use of fractals in image encryption.

For keystream generation, fractal geometry and chaotic maps have been extensively used in both cryptography and data hiding [8]. As for the use of chaotic function, Liu and Wang [9] used a piecewise linear chaotic map to generate a pseudorandom key stream sequence. Amin et al. [10] used a chaotic block cipher scheme to encrypt a block of bits rather than a block of pixels using cryptographic primitive operations and a nonlinear transformation function. Wang et al. in [11] encrypted images using Baker map and several one-dimensional chaotic maps. Wang et al. [12] encrypted plaintext by alternating between stream ciphering and block ciphering based on a pseudorandom number generated based on a chaotic map. On the other hand, Huang [13] used a nonlinear chaotic Chebyshev function to generate a keystream, in addition to multiple pixel permutations. In [14], Wang and Luan proposed a novel image encryption scheme based on reversible cellular automata combined with chaos.

As for the use of fractal images, the Mandelbulb set has been used in [15]. The fractal image and the fractal-compressed source image are transformed to square matrices and matrix operations are applied in encryption and decryption. The Mandelbrot set is utilized in [16] along with the Hilbert transformation to generate a random encryption key. Moreover, Abd-El-Hafiz et al. [6] introduced a novel image encryption system based on diffusion and confusion processes in which the image information is hidden inside the complex details of multiple fractal images.

The FFCT was first defined in [19] and corresponds to a finite field version of the discrete cosine transform (DCT). It exhibits interesting properties which are valuable for cryptographic purposes. In [5], a simple method for uniformizing histograms of greyscale digital images was introduced based on 8-point FFCT without any encryption mechanism. Then, a method for histogram uniformization of greyscale images integrated with a full encryption mechanism was developed in [17]. In [20], an improvement had been proposed to allow the color channels to be processed jointly by means of a single transformation round by using a 32-point FFCT over the $\mathrm{GF}(2^{24})$ to transform blocks of the image.

The present work is based on the idea of multiphase image encryption and thus the encryption process is split over two phases: FFCT phase and encryption phase as in [18]. First, a transformation is done based on applying a recursive 8-point FFCT over $\mathrm{GF}(2^8)$ to blocks of a color image. Since RGB images are considered, the transform is applied to each color channel separately. Application of this transformation guarantees a uniform histogram as well as decorrelation of values of adjacent pixels. This makes our method useful to provide robustness against statistical attacks in image encryption schemes. Second, encryption is based on using fractal images to generate the keystream which leads to a simple, yet secure encryption scheme.

3. Background

In this section, some important definitions are provided which are essential to the development of the proposed scheme. First, the finite field cosine transform (FFCT) is defined, and then zigzag image scanning and fractal images and their generation are described.

3.1. Finite Field Cosine Transform. In order to define the FFCT as in [5], let λ be a nonzero element in the finite field $\mathrm{GF}(p)$, where p is an odd prime. The finite field cosine function related to λ is computed modulo p as

$$\cos_\lambda(x) \coloneqq \frac{\lambda^x + \lambda^{-x}}{2}, \quad x = 0, 1, \ldots, \mathrm{ord}(\lambda), \tag{1}$$

where $\mathrm{ord}(\lambda)$ denotes the multiplicative order of λ.

As an advantage, the finite field cosine transform is defined over the integers modulo p and thus avoids inaccuracies associated with other real-valued transforms. A type-2 FFCT is defined in [5, 19] as follows.

Let $\lambda \in \mathrm{GF}(p)$ be an element with multiplicative order $2N$. The finite field cosine transform of the vector $x = [x_0, x_1, \ldots, x_{N-1}]$, $x_i \in \mathrm{GF}(p)$ is given by the vector $X = [X_0, X_1, \ldots, X_{N-1}]$, $X_k \in \mathrm{GF}(p)$ whose elements are

$$X_k \coloneqq \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \delta_k x_i \cos_\lambda\left(k \frac{2i+1}{2}\right), \tag{2}$$

where

$$\delta_k = \begin{cases} \dfrac{1}{\sqrt{2}}, & k = 0 \\ 1, & k = 1, 2, \ldots, N-1 \end{cases}. \tag{3}$$

The inverse FFCT can be computed according to the following formula:

$$x_i \coloneqq \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \delta_i X_k \cos_\lambda \left(k \frac{1}{2} i\right). \tag{4}$$

The previous equations can be written in matrix format as $X = Tx$, where T corresponds to the transform matrix, the elements of which are obtained directly from (2). Using such a matrix notation, the FFCT can be extended to two dimensions. The two-dimensional FFCT of a matrix M with dimensions $N \times N$ can be computed by

$$C = TMT^t \pmod{p}, \tag{5}$$

where T is the transformation matrix.

In this paper, RGB color images are considered, which have three color channels red, green, and blue. Each channel has pixels values ranging from 0 to 255. As suggested by [5], the Fermat prime used in the current paper is taken to be $p = 257$. An example of a 8×8 transform matrix used for testing the scheme is the following:

$$T = \begin{bmatrix} 15 & 15 & 15 & 15 & 15 & 15 & 15 & 15 \\ 137 & 163 & 98 & 106 & 151 & 159 & 94 & 120 \\ 160 & 6 & 251 & 97 & 97 & 251 & 6 & 160 \\ 163 & 151 & 120 & 159 & 98 & 137 & 106 & 94 \\ 242 & 15 & 15 & 242 & 242 & 15 & 15 & 242 \\ 98 & 120 & 106 & 163 & 94 & 151 & 137 & 159 \\ 6 & 97 & 160 & 251 & 251 & 160 & 97 & 6 \\ 106 & 159 & 163 & 120 & 137 & 94 & 98 & 151 \end{bmatrix}, \tag{6}$$

where T is an orthogonal matrix which satisfies

$$TT^t = T^t T = I_N. \tag{7}$$

The period of the FFCT transform matrix T has great importance for the application described in this paper, since such a parameter corresponds to the least integer and positive power l giving $T^l = I$. Once the transform matrix T used has a large period l ($l = 16974594$ for the above matrix), there is not risk of returning the transformed block to its original block by the FFCT repetitive computation for a small number of times.

3.2. Zigzag Scanning of Images and Reshaping. To understand the idea of zigzag scanning of a two-dimensional matrix, see Figure 1. A two-dimensional matrix is scanned in a zigzag way to obtain a one-dimensional vector. The one-dimensional vector thus has N^2 elements, which is then to be reshaped back into an $N \times N$ two-dimensional matrix. Every N consecutive element is used to form a column of this matrix as shown in Figure 2. This method of image scanning is used in the keystream generation step mentioned in Section 5.2.

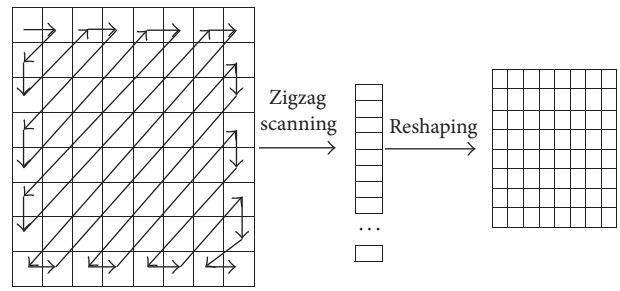


Figure 1: Zigzag scanning of image.

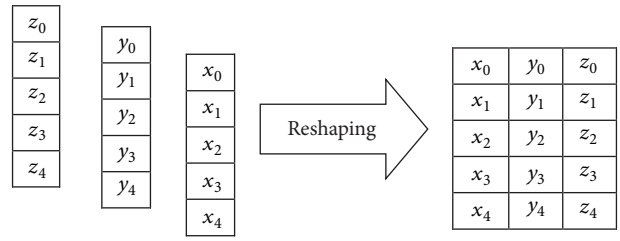


Figure 2: Reshaping 1D vector into 2D matrix.

3.3. Fractal Images. A fractal object [8] is a self-similar object obtained by repeating a kernel at various scales of magnification and can possess high variations. A fractal image can be generated using a mathematical equation or function that is iterated for a finite number of times. The choice of a fractal image is flexible and it depends on the level of detail and colors. Several resources are available on the Internet to generate fractals.

Many of the most popular fractal images can be obtained through Iterated Function Systems (IFS) [8]. IFS have received a lot of attention because of their appealing combination of conceptual simplicity, computational efficiency, and great ability to reproduce natural formations and complex phenomena [21]. There are miscellaneous programs, which are freely available on the Internet, for generating and rendering IFS fractals. Figure 3 shows a sample of eight fractals used in the testing phase of our scheme.

4. Encryption Evaluation Techniques

There are two main categories for performance evaluation techniques: statistical measures and sensitivity measures. In this section, the evaluation techniques used in the assessment of the performance of the proposed encryption scheme are reviewed.

4.1. Statistical Measures. The statistical means include measuring the correlation among the image pixels, histogram analysis, entropy analysis, and the NIST suite of randomness tests.

Adjacent image pixels in a plain image are highly correlated with each other and, hence, one of the encryption targets is to reduce the *correlation coefficients* for horizontal, vertical, and diagonal pixels. The correlation coefficient (ρ) between

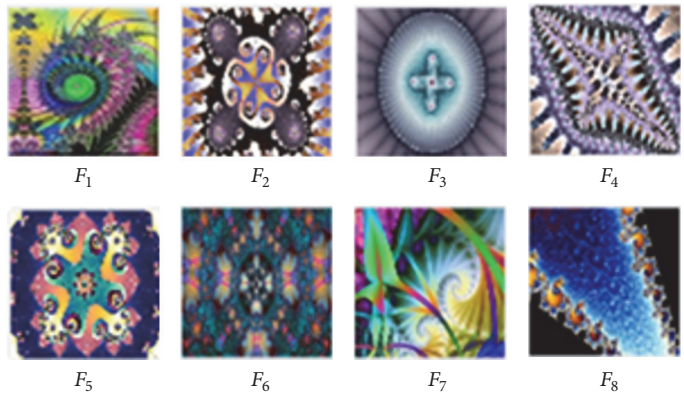


Figure 3: Eight fractal images used in testing the proposed system.

two N-dimensional vectors x and y is calculated using the following formula [4]:

$$\begin{aligned}&\mathrm{Cov}(x, y)\\ &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{j=1}^{N}x_j\right)\left(y_i - \frac{1}{N}\sum_{j=1}^{N}y_j\right),\end{aligned} \tag{8a}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{j=1}^{N}x_j\right)^2, \tag{8b}$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \frac{1}{N}\sum_{j=1}^{N}y_j\right)^2, \tag{8c}$$

$$\rho = \frac{\mathrm{Cov}(x, y)}{\sqrt{D(x)\,D(y)}}. \tag{8d}$$

Histogram analysis is a visual test that shows the distribution of pixel color values across the whole image. For normal images, the histogram shows curves and peaks which means that some specific color values appear more than others. On the other hand, an encrypted image histogram should be flat as no specific color value should appear more than another color value.

Entropy is a measure of the predictability of a random source. Owing to the high correlation between adjacent pixels, image data is predictable and consequently has low entropy. Encrypted image data, on the other hand, should appear random to avoid any information leakage; that is, all pixel values are equally likely to occur. For a binary source producing 2^8 symbols of equal probabilities and each symbol is 8 bits long, the entropy of this source is defined as

$$\mathrm{Entropy} = -\sum_{i=1}^{2^8} p(s_i)\log_2\left(p(s_i)\right). \tag{9}$$

This source is considered unpredictable for an entropy value of 8.

NIST SP-800-22 statistical test suite was introduced by [4], which is a group of 15 different tests designed to assess the random characteristics of a group of bits. Such bits can be the output of a random number generator or pixels of an encrypted image. The NIST tests in the second scenario examine the effectiveness of the encryption technique showing how similar the encrypted image is to a random noise based on the P value (PV) of the test and the proportion of passing sequences (PP).

4.2. Sensitivity Measures. Sensitivity tests are used in investigating the sensitivity of cryptosystems to changes in plaintext and system key. Secure cryptosystems should be very sensitive to small changes in either plaintext or system key. Sensitivity tests include differential attack measures, mean square error, and one bit change in system key.

4.2.1. Resistance against Differential Attacks. These study the statistical characteristics of the input plain image and the output enciphered image with the aim to infer any meaningful relationships when changing the input while observing the output to reveal the encryption algorithm. A strong encryption algorithm should be sensitive in the sense that small changes to the input produce a significant change in the output. Quantitatively, different measures are defined for evaluating the protection levels against differential attacks [22].

The mean absolute error (MAE) measures the absolute change between the encrypted image E and the source image P. Let W and H be the width and height of the source image, respectively; then

$$\mathrm{MAE} = \frac{1}{W \times H}\sum_{i=1}^{H}\sum_{j=1}^{W}\left|P(i, j) - E(i, j)\right|, \tag{10}$$

where $P(i, j)$ is the original pixel value at location (i, j) and $E(i, j)$ is the encrypted pixel value at the same location.

The number of pixels change rate (NPCR) is used to measure the percentage of different pixels between two encrypted images E_1 and E_2 whose corresponding two original images

are identical except for only one-pixel difference and it is calculated using

$$D(i,j) = \begin{cases} 0 : E_1(i,j) = E_2(i,j) \\ 1 : E_1(i,j) \neq E_2(i,j), \end{cases} \quad (11a)$$

$$\text{NPCR} = \frac{1}{W \times H} \sum_{i=1}^{H} \sum_{j=1}^{W} D(i,j) \times 100\%. \quad (11b)$$

The unified average changing intensity (UACI) measures the average intensity of differences between two encrypted images provided that their two corresponding original images are identical except for only one-pixel difference and it is calculated as

$$\text{UACI} = \frac{1}{W \times H} \sum_{i=1}^{H} \sum_{j=1}^{W} \left| \frac{E_1(i,j) - E_2(i,j)}{255} \right| \times 100\%. \quad (12)$$

4.2.2. Mean Square Error. The mean square error value gives another indication on how far the encrypted image is from the original image. As this value gets larger this implies a better encrypted image. The mean square error is calculated using the following equation:

$$\text{MSE} = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} [P(i,j) - E(i,j)]^2}{W \times H}. \quad (13)$$

Recommended values, as reported by [22], of MSE are in the order of 10^4 for (1024×1024) images.

4.2.3. One Bit Change in System Key. This test is used to test and measure the sensitivity of the system key to one bit change. The system key contains the values of the system parameters that are used to initialize the encryption process. A secure encryption process is sensitive to any slight change in any of its parameters and, hence, one bit change in the system key should lead to a totally different behavior in the encryption process. If a system is not sensitive enough to one bit change in the system key, this would lead to revealing the encryption without knowing the exact secret key, which is not acceptable in any practical cryptosystem. For one bit change in system key, two different tests are performed.

Test I. It is changing one bit in the fractals count part (S) of the key, which leads to the selection of a different number of fractals.

Test II. It is changing one bit in the fractals number part of the key, which leads to the selection of one different fractal other than the originally selected one.

4.3. Security Analysis. Security of the proposed scheme is analyzed by investigating its resistance to classical cryptographic attacks, like known-plaintext and chosen-plaintext attacks to reveal the secret parameters of the algorithm.

In *known-plaintext attack*, the attacker knows at least one sample of both the plaintext and the ciphertext. In this case, the attacker can use that to break algorithm.

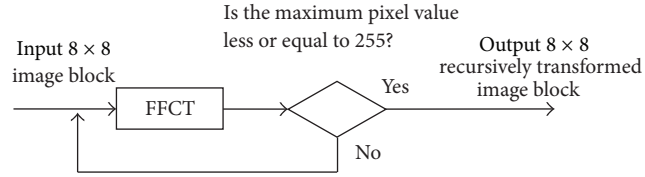


Figure 4: Recursive computation of the FFCT of an 8 × 8 image block [17].

In *chosen-plaintext attack*, the attacker can specify his own plaintext and encrypt it and then uses the result to determine the encryption key.

5. The Proposed Encryption Scheme

First, the FFCT algorithm used in the first phase is described. Next, the generation of the keystream based on multiple fractal images is described. Finally, the pseudocode for the proposed encryption algorithm is provided.

5.1. FFCT Phase. The three RGB component matrices of the source color image are separated. Each component matrix is divided into 8 × 8 blocks. The application of an agreed upon transform matrix T to each 8 × 8 image block produces a matrix (transformed version of the image block), where the elements can range from 0 to 256; this is due to the fact that modulo 257 arithmetic is being used. If any element in a transformed block is equal to 256, such a block cannot be coded using 8 bits per pixel.

The technique adopted by [17] calls for the recursive application of the transform to overcome this problem; that is, the FFCT of a block is computed repeatedly until the resulting block has no pixels with value equal to 256; see Figure 4. This process is invertible by computing the inverse transform also repeatedly until encountering the original image block, which does not contain any pixel with value equal to 256. This technique is suggested and applied in [5, 17] to encrypt greyscale images. In this paper, this technique is extended to encrypt color images by separating and treating each one of the three RGB color channels independently.

The original image block can be restored by applying the inverse finite field cosine transform (IFFCT); IFFCT of a matrix C with dimensions $N \times N$ can be computed as $M = T^{-1}C(T^t)^{-1}$ (modulo p). If T is an orthogonal matrix as in (6), its inverse is simply its transpose. This adds to the simplicity and computational efficiency of the algorithm. Another interesting characteristic of the above FFCT transform computational algorithm is that it lends itself to parallel implementation, where processing of all blocks can be done in parallel.

5.2. Pseudorandom Keystream Generation Using Fractals. Practically, assume that 2^K fractal images $F_1, F_2, \ldots F_{2^K}$ are available to both the sender and receiver, and only a subset of S fractal images are used in keystream generation, where $S \leq 2^K$. Hence, the number of fractal images used is a variable part

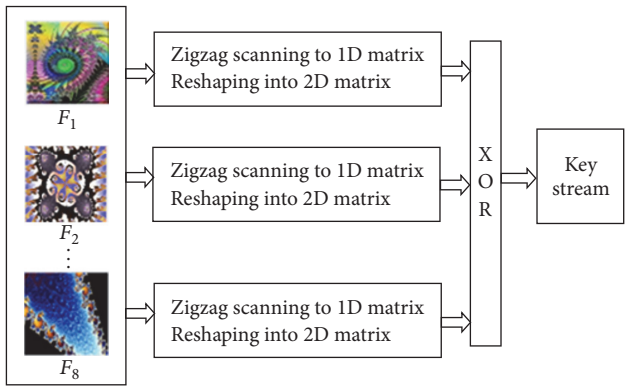


Figure 5: Example of key stream generation using 8 fractals.

of the system key. It is noteworthy that the selected fractals should have the same resolution as the plain image.

To generate the keystream to be used in the enciphering phase, a new method is suggested using multiple fractal images rather than the one mentioned in [6]. As shown in Figure 5, the keystream is generated by zigzag scanning each fractal image of a variable number of fractals S and then reshaping each into a matrix and finally XORing the resulting reshaped fractals. A sample of fractals used ($S = 8$) in our work is shown in Figure 3.

5.3. Pseudocode for the Proposed Scheme. Suppose a sender A wants to encrypt an image IM and send it to its recipient B; then A carries out the following steps, assuming that both A and B have agreed on the fractals to be used in the generation of the keystream as well as the transformation matrix T.

5.3.1. Encryption Process

Phase 1 (FFCT Phase)

Step 1. From the $1024 \times 1024 \times 3$ input image, extract the three red, green, and blue color channels.

Step 2. For each channel, divide the 1024×1024 image channel into 16384 blocks with each block being 8×8 in size.

Step 3. For each 8×8 block, apply FFCT recursively as in Figure 4.

Step 4. Group the transformed blocks to get the intermediate image.

Phase 2 (Encryption Phase)

Step 5 (key generation). Generate the keystream using the selected fractals agreed upon, by zigzag scanning each fractal and reshaping it and finally XORing the resulting reshaped eight fractals.

Step 6. Encrypt the image by bit XORing the intermediate image resulting from Step 3 with the keystream of Step 5.

The block diagram of the proposed encryption process is shown in Figure 6.

5.3.2. Decryption Process. The proposed decryption process proceeds as follows, with the block diagram depicted in Figure 7.

Step 1. Perform bit XORing between the ciphered image and the private keystream generated from selected fractals agreed upon to get the intermediate color image.

Step 2. From the $1024 \times 1024 \times 3$ intermediate color image, extract the three red, green, and blue color channels.

Step 3. For each channel, divide the 1024×1024 image channel into 8×8 blocks.

Step 4. For each 8×8 block, apply the IFFCT as mentioned in Section 5.1.

Step 5. Reconstruct the plain color image from 8×8 blocks of each color channel.

6. System Key Analysis

The system key is composed of the parameters, which are used in the (encryption/decryption) process. It includes the following parameters:

(i) For FFCT phase, consider the 8×8 transformation matrix T of unsigned 8-bit integers (0 to $2^8 - 1$), which is used for all 8×8 image blocks. It can be generated by specifying λ in (1); thus only 8 bits are required for generating the matrix T from λ.

(ii) K bits are required to specify the number of selected fractals, where 2^K is the maximum number of available fractals.

(iii) Consider $S \times K$ bits to specify the selected fractals' numbers, where S is the actual number of fractals selected and where $S \leq 2^K$.

Thus, the key length is variable depending on number of fractals available 2^K and the number of selected fractals S and equals $8 + K(1 + S)$ bits. In our experiment, 8 fractals are selected from 2^7 available fractals ($S = 8$, $K = 7$); hence key length used in our experiment equals 71 bits.

7. Computational Complexity Analysis

The computational complexity of the proposed encryption scheme was calculated for the two phases:

(i) For the FFCT phase, we used 8×8 matrix multiplication. Since the image resolution is $R \times R$, the number of blocks is $(R \times R)/(8 \times 8)$. Thus, the complexity of this step is $O(R^2)$.

(ii) For the encryption phase, the complexity of zigzag scanning of fractals and XORing with the source image are known to be $O(R^2)$.

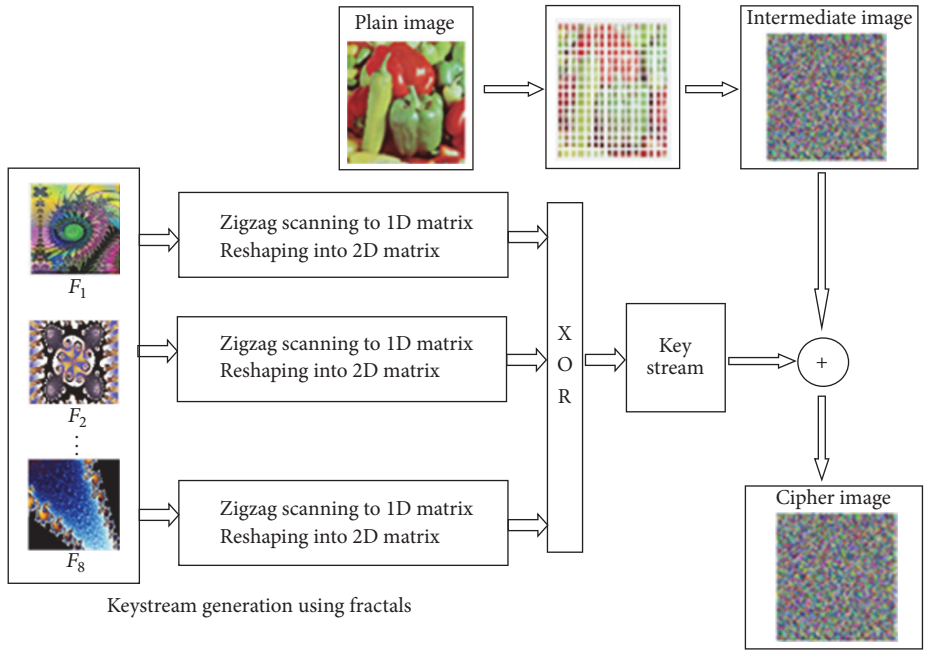


Figure 6: Block diagram of encryption process.

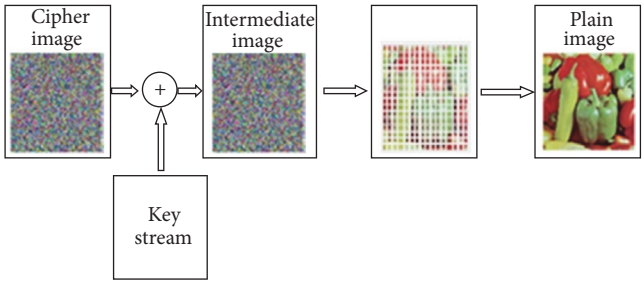


Figure 7: Block diagram of decryption process.

Accordingly, the computational complexity of the proposed encryption/decryption scheme is $O(R^2)$, where $R \times R$ is the source image resolution. Moreover, the computational complexity can be improved by using fast algorithms for computing the FFCT by $O(R \log R)$ as in [23].

8. Speed Analysis

After running the proposed scheme many times on different plain images, the approximate time necessary to perform encryption/decryption of a 1024×1024 colored image was 7 seconds; thus the average speed of our proposed scheme is approximately 0.45 MB/second. These results were obtained using an HP 250 G3 computer equipped with an Intel(R) Core (TM) i3-4005U CPU @1.70 GHz processor and 4 GB of RAM running Windows 10 64-bits. The block-by-block FFCT computation makes it possible to reduce the time required by the proposed scheme if parallel processing is employed. Moreover, there are fast algorithms for computing the FFCT as in [23].

9. Experimental Results

All the work in this paper is implemented using MATLAB. We apply our MATLAB program on 1024×1024 standard images available from [24]. In order to evaluate the performance of the proposed scheme, our encryption scheme has been applied to some standard images, for example, Mandrill (4.2.03), airplane (4.2.05), and peppers (4.2.07). For the fractals, 2^K Mandelbulb fractals can be readily available (from the Internet) and S fractals are chosen.

The encryption quality is evaluated using different analysis techniques including both statistical means like correlation coefficients, histogram analysis, entropy analysis, and NIST test suite. Moreover, sensitivity tests like differential attack measures, MSE, and one bit change in system key have been investigated.

It is clear from the results that the proposed scheme successfully meets various requirements for a strong image encryption scheme as detailed in the following points.

(i) As apparent in Figures 8, 9, and 10, the histogram of ciphered images is uniform with all pixel values being equally likely in the three color channels.

(ii) In Figure 11, the correlation among pixels has decreased in the encrypted image and the correlation coefficients are approximately equal to zero.

(iii) Moreover, in Figure 11, the entropy is shown to increase after applying the proposed encryption algorithm and its value approaches 8 (the ideal value).

(iv) Tables 1, 2, and 3 show that the encrypted images successfully passed the NIST suite tests. The symbol N\A in these tables means not available; that is, testing randomness according to this criterion failed.

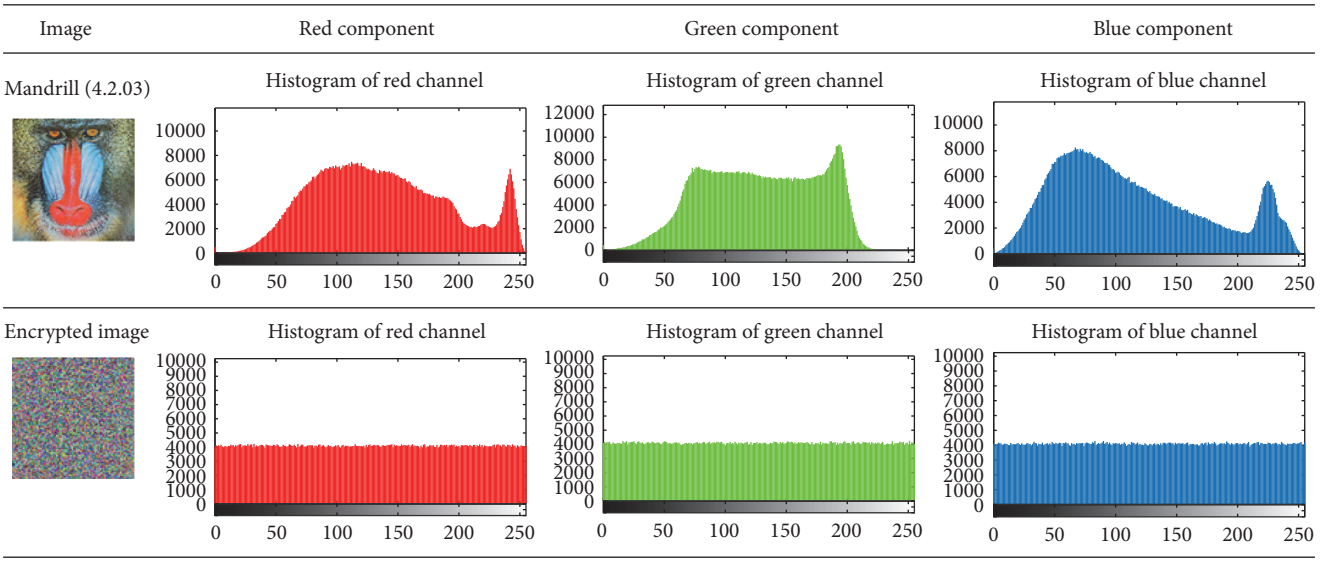


Figure 8: Histogram analysis of Mandrill original and encrypted image.

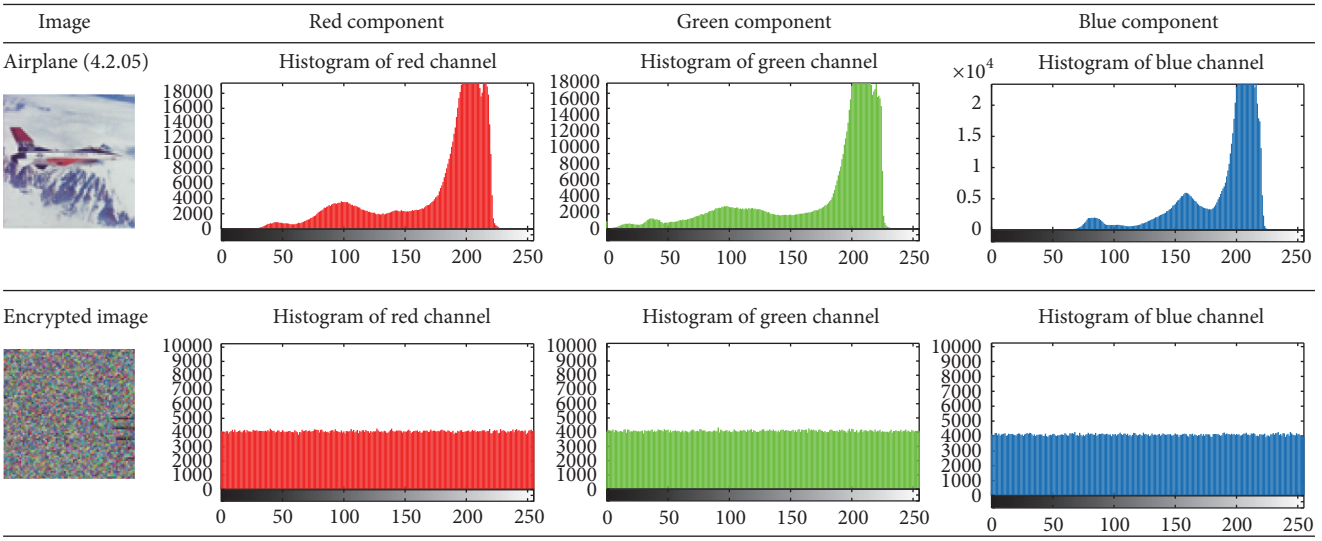


Figure 9: Histogram analysis of airplane original and encrypted image.

(v) From Figure 12, it is apparent that the proposed scheme is resistant to differential attacks. This is clear since the scheme is sensitive to only one pixel change in the plain image as indicated by the values of MAE, NPCR, and UACI measures.

(vi) As for MSE, from Figure 12, it is apparent that the mean square error between the original image and the encrypted image is large enough in the order of 10^4 as recommended for 1024×1024 image.

(vii) As for key sensitivity, Table 4 shows the results of the two tests. NPCR% is computed between the original plain image and the image decrypted using a wrong key in the two tests. As NCPR% approaches 100%, it is clear that the system is very sensitive to key changes.

(viii) Figure 13 shows that the encryption result is very sensitive to a change in the key, regardless of where the change occurs. All of the tests give a totally wrong image other than the peppers image.

10. Security Analysis

In this section, we analyze the security of the proposed scheme. We mount both known-plaintext attack and chosen-plaintext attack and attempt to reveal the secret parameters of the algorithm. However, it appears that the proposed scheme is resistant to both attacks.

10.1. Known-Plaintext Attack. Assume that the plain image-cipher image pair (M, C) is known to the attacker. Hence,

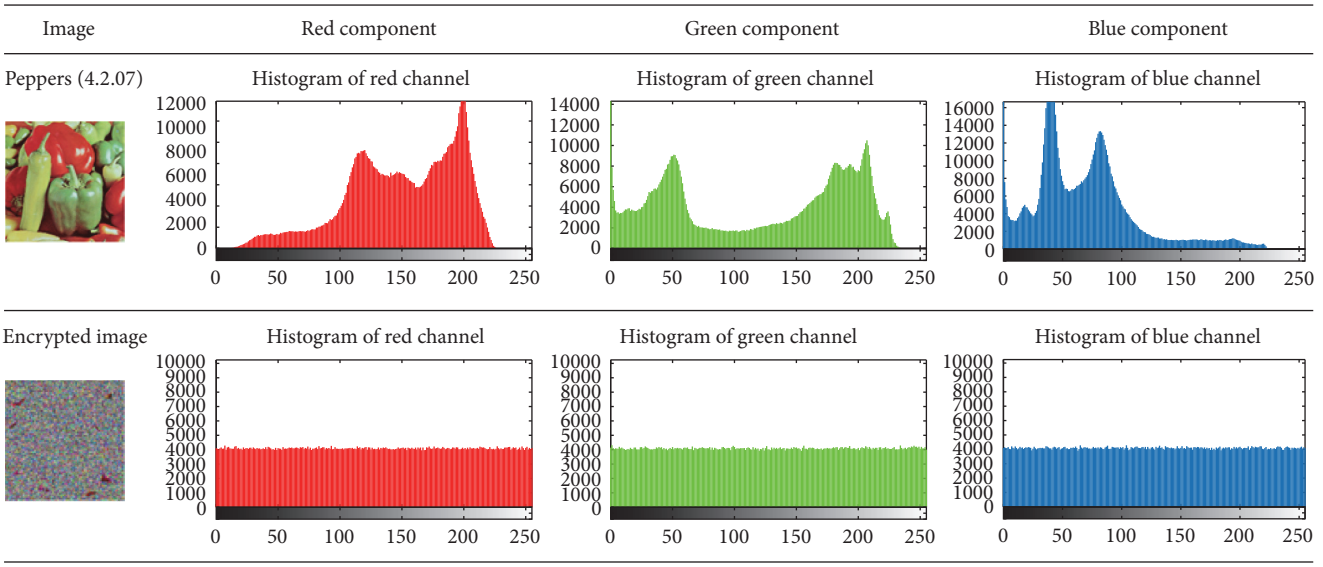


Figure 10: Histogram analysis of peppers original and encrypted image.

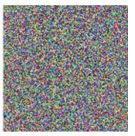
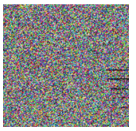
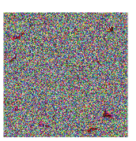
Original image		Pixel correlation coefficients			Entropy	Encrypted image		Correlation coefficients			Entropy
		Horz.	Vert.	Diag.				Horz.	Vert.	Diag.	
	R	0.9828	0.9675	0.9549	7.6798		R	0.0013	0.0006	0.0008	7.9998
	G	0.9682	0.9408	0.9176	7.4469		G	0.0001	0.0006	0.0001	7.9998
	B	0.9776	0.9706	0.9513	7.7344		B	0.001	0.0004	0.0012	7.9998
	Avg.	0.9762	0.9597	0.9413			Avg.	0.0008	0.0006	0.0007	
	R	0.9930	0.9890	0.9824	6.7241		R	0.0029	0.0001	0.0021	7.9998
	G	0.9895	0.9918	0.9821	6.8103		G	0.0050	0.0006	0.0002	7.9999
	B	0.9910	0.9838	0.9767	6.2166		B	0.0017	0.0015	0.0003	7.9998
	Avg.	0.9912	0.9882	0.9804			Avg.	0.0032	0.0007	0.0009	
	R	0.9922	0.9929	0.9865	7.3255		R	0.0009	0.0000	0.0012	7.9998
	G	0.9956	0.9957	0.9914	7.5324		G	0.0091	0.0079	0.0058	7.9998
	B	0.9922	0.9921	0.9853	7.0896		B	0.0061	0.0059	0.0045	7.9998
	Avg.	0.9933	0.9936	0.9877			Avg.	0.0053	0.0046	0.0039	

Figure 11: Correlation coefficients and entropy values of some 1024 × 1024 original and encrypted test images.

Table 1: NIST test results for Mandrill and encrypted Mandrill.

Test	Mandrill		Encrypted Mandrill	
	PV	PP	PV	PP
Frequency	×	0.42	√	1.000
Block frequency	×	0.000	√	1.000
Cumulative sums	×	0.750	√	1.000
Runs	×	0.250	√	1.000
Longest run	×	0.000	√	1.000
Rank	×	0.000	√	1.000
FFT	×	0.000	√	1.000
Nonover. template	×	0.000	√	0.958
Overlapping template	×	0.000	√	1.000
Universal	×	0.000	√	1.000
Approximate entropy	×	0.000	√	1.000
Random excursion	N\A	N\A	√	1.000
Ran. excursion variant	N\A	N\A	√	1.000
Serial	×	0.000	√	1.000
Linear complexity	√	1.000	√	1.000
Final result	Fail		Success	

Table 2: NIST test results for airplane and encrypted airplane.

Test	Airplane		Encrypted airplane	
	PV	PP	PV	PP
Frequency	×	0.167	√	1.000
Block frequency	×	0.000	√	1.000
Cumulative sums	×	0.125	√	1.000
Runs	×	0.042	√	1.000
Longest run	×	0.000	√	1.000
Rank	×	0.000	√	1.000
FFT	×	0.000	√	1.000
Nonover. template	×	0.000	√	0.967
Overlapping template	×	0.000	√	1.000
Universal	×	0.000	√	1.000
Approximate entropy	×	0.000	√	1.000
Random excursion	N\A	N\A	√	1.000
Ran. excursion variant	N\A	N\A	√	1.000
Serial	×	0.000	√	1.000
Linear complexity	√	1.000	√	1.000
Final result	Fail		Success	

for each 8×8 block m, $c = TmT^t \oplus F_1$, where T is the transformation matrix and F_1 is the associated block of the generated keystream by the fractals selected. Thus, going from C to M for another pair is not such an easy operation; the attacker should know the transformation matrix T and also should know number of fractals used S as well as which fractals are used. Even if the attacker knows more plain image-cipher image pairs, he could not find out the system key because the keystream is varied every run. So, the proposed scheme is robust against known-plaintext attack.

Table 3: NIST test results for peppers and encrypted peppers.

Test	Peppers		Encrypted peppers	
	PV	PP	PV	PP
Frequency	×	0.50	√	1.000
Block frequency	×	0.000	√	0.958
Cumulative sums	×	0.542	√	0.979
Runs	×	0.458	√	1.000
Longest run	×	0.000	√	1.000
Rank	×	0.000	√	1.000
FFT	×	0.000	√	1.000
Nonover. template	×	0.000	√	0.990
Overlapping template	×	0.000	√	1.000
Universal	×	0.000	√	0.958
Approximate entropy	×	0.000	√	0.997
Random excursion	N\A	N\A	√	1.000
Ran. excursion variant	N\A	N\A	√	0.997
Serial	×	0.000	√	0.959
Linear complexity	√	1.000	√	1.000
Final result	Fail		Success	

Table 4: Wrong decryption for the peppers image (NPCR %).

Test	NPCR %		
	R	G	B
Test I	99.79	99.74	99.77
Test II	99.79	99.75	99.78

10.2. Chosen-Plaintext Attack. Assume the attacker chooses an all zero plain image M_1 and observes the ciphertext C_1. Hence, it is easy to verify that the ciphertext $C_1 = F$, which is the keystream itself.

Again, if the attacker chooses another plain image M, hence, for each 8×8 block m, $c = TmT^t \oplus F^*$. If the same keystream is reused, the attacker can find out $T^{-1}m(T^t)^{-1}$. However, he cannot easily find out the matrix T, especially with the recursive application of the transform to various plain image blocks. But still, we suggest that the keystream is changed every time a connection is initiated between the sender and receiver; that is, we use a one-time-pad system to secure the proposed scheme against the above attack.

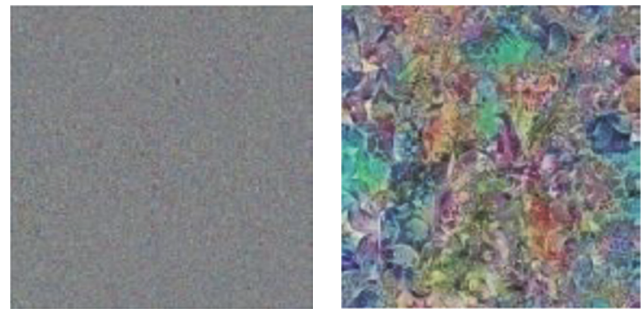
11. Discussion

The proposed two-phase image encryption scheme benefits from the histogram equalization capabilities of the FFCT and blends it with the success of fractal images in producing highly random keystreams. The FFCT can be computed in parallel for all 8×8 blocks of the image for efficiency. The resulting cipher image appears as random noise as clear from previous histogram analysis and application of NIST suite tests.

In our simulations, the number of recursive applications of FFCT to an image block has been investigated and the

Simulation result			Differential attacks measures			MSE
				MAE	NPCR%	UACI%	
Original	Encrypted	Decrypted	R	76.0048	99.6617	33.4502	8651
			G	72.4362	99.6370	33.4866	7728
			B	79.2610	99.6235	33.4524	9508
			Avg.	75.90067	99.6407	33.4637	8629
Original	Encrypted	Decrypted	R	81.5138	99.5758	33.4640	9955
			G	84.3053	99.5449	33.4448	10670
			B	83.3591	99.5813	33.4785	10394
			Avg.	83.0594	99.567	33.4624	10340
Original	Encrypted	Decrypted	R	73.9011	99.6917	33.4720	7978
			G	86.4335	99.6500	33.4975	11208
			B	86.0157	99.7103	33.5152	11109
			Avg.	82.1168	99.684	33.4949	10098

Figure 12: Differential attacks and MSE of some encrypted images.



(a) (b)

Figure 13: Wrong decryption of peppers image: (a) test I and (b) test II.

percentage of image blocks being subjected to a given number of recursive applications of the FFCT has been computed. Such results are shown in Table 5. In all images, for each color channel almost 80% of the blocks had to be transformed only once. A rather small percentage of blocks had to be transformed more than twice. This indicates that the extra computational effort due to the recursive application of the FFCT is minor. It has also been observed that the largest number of rounds necessary for a block was 8. An appropriate choice of the transformation matrix T should be one with a large period l (in our experiments $l = 16974594$). Thus, there was no risk of returning the transformed block to its original block by the FFCT repetitive application.

In order to further assess the success of the proposed scheme, its performance is compared to several other schemes in literature. Table 6 shows the results obtained when encrypting the Lena (512×512) image using the proposed encryption scheme and compares its performance

Table 5: Average percentages of RGB channel blocks of test images submitted to recursive applications of the FFCT.

FFCT number of applications	Mandrill	Airplane	Peppers
1	78.2726	77.7355	78.1487
2	17.3096	17.2607	17.5293
3	3.3447	3.9795	3.4180
4	0.7812	0.8057	0.6348
5	0.2197	0.1465	0.1953
6	0.0488	0.0732	0.0488
7	0.0244	0.0000	0.0000
8	0.0000	0.0000	0.0000

Table 6: Comparison among the proposed scheme and other schemes in literature.

Researches	Pixel correlation			NPCR%	UACI%	Entropy
	Horz.	Vert.	Diag.			
This work	0.00069	0.0007	0.0002	99.7248	33.4647	7.9999
[6]	0.0021	0.0009	0.0018	99.740	33.470	7.9997
[9]	0.0965	0.0318	0.0362	99.633	33.458	7.9845
[10]	0.0209	0.0144	0.035	99.610	33.410	7.9998
[11]	0.0141	0.0107	0.0097	99.670	27.880	—
[12]	0.0140	0.0092	0.0051	98.563	33.081	7.9940
[13]	0.0974	0.0707	0.0484	99.684	33.439	—
[14]	0.0011	0.0193	0.0045	99.790	33.350	7.9992
[17]	−0.0049	0.0015	0.0021	99.6066	33.4758	—
[18]	0.00089	0.00170	—	—	—	7.9993

with other related recent schemes. The comparison involves correlation coefficients, some differential attacks measures, and the entropy.

Like our scheme, a colored image is encrypted in [6, 9, 10], while in other researches greyscale images are only considered [11, 12, 17, 18]. In Table 6, for a fair comparison, the averages of the performance measures for the proposed scheme for the three RGB channels are compared against the averages of other schemes. This is how we compared our technique using colored Lena image with some other techniques using a greyscale Lena image. The obtained results show great potential compared to other techniques.

12. Conclusion

In this paper, an efficient scheme for colored image encryption using image pixel value transformation is proposed. The scheme employs the finite field cosine transform FFCT to reduce correlation among adjacent pixels and to obtain a uniform histogram for the enciphered images. The FFCT does not involve any approximations inherent in other transforms as it operates on integers modulo p and results in integer sequences. Moreover, encryption is performed on the transformed image using multiple fractals to enforce more security. The experimental results show that the cipher image has entropy information close to ideal value 8 and low correlation coefficients close to ideal value 0. Thus, the analysis proves the security, correctness, effectiveness, and robustness of the proposed image encryption algorithm. Additionally, the number of recursive applications of the FFCT to an image block has been investigated and the experiments showed that it has minor effect on the algorithm computational efficiency as it is in 80% of the cases equal to one and the FFCT algorithm can proceed on all blocks in parallel. Finally, for the scheme to resist both known-plaintext and chosen ciphertext attacks, it is suggested to refresh the keystream used in the enciphering phase per communication session and to keep the transformation matrix used in the FFCT phase as a shared secret between the communicating parties.

Competing Interests

The authors declare that they have no competing interests.

References

[1] C. Li, *Cryptanalyses of some multimedia encryption schemes [M.S. dissertation]*, Zhejiang University, Hangzhou, China, 2005.

[2] W. Stallings, *Cryptography and Network Security: Principlesand Practices*, Pearson Education India, 2006.

[3] N. Rawal and M. Dhawan, "A survey report on imageencryption techniques," *International Journal of Engineering Research and Technology*, vol. 2, no. 10, 2013.

[4] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, *A Statistical Test Suite for Random and Pseudorandom Number*

Generators for Cryptographic Applications, Booz-Allen and Hamilton Inc, Mclean, Va, USA, 2001.

[5] J. B. Lima and R. M. C. de Souza, "Histogram uniformization for digital image encryption," in *Proceedings of the 25th Conference on Graphics, Patterns and Images (SIBGRAPI '12)*, pp. 55–62, IEEE, Ouro Preto, Brazil, August 2012.

[6] S. K. Abd-El-Hafiz, A. G. Radwan, S. H. Abdel Haleem, and M. L. Barakat, "A fractal-based image encryption system," *IET Image Processing*, vol. 8, no. 12, pp. 742–752, 2014.

[7] K. D. Patel and S. Belani, "Image encryption using differenttechniques: a review," *International Journal of Emerging Technology and Advanced Engineering*, vol. 1, no. 1, pp. 30–34, 2011.

[8] P. S. Addison, *Fractals and Chaos: An Illustrated Course*, CRC Press, 1997.

[9] H. Liu and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Computers & Mathematics with Applications*, vol. 59, no. 10, pp. 3320–3327, 2010.

[10] M. Amin, O. S. Faragallah, and A. A. Abd El-Latif, "A chaotic block cipher algorithm for image cryptosystems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3484–3497, 2010.

[11] X.-Y. Wang, F. Chen, and T. Wang, "A new compound mode of confusion and diffusion for block encryption of image based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2479–2485, 2010.

[12] X. Wang, X. Wang, J. Zhao, and Z. Zhang, "Chaotic encryption algorithm based on alternant of stream cipher and block cipher," *Nonlinear Dynamics. An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems*, vol. 63, no. 4, pp. 587–597, 2011.

[13] X. Huang, "Image encryption algorithm using chaotic Chebyshev generator," *Nonlinear Dynamics*, vol. 67, no. 4, pp. 2411–2417, 2012.

[14] X. Wang and D. Luan, "A novel image encryption algorithm using chaos and reversible cellular automata," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 11, pp. 3075–3085, 2013.

[15] A. J. J. Lock, C. H. Loh, S. H. Juhari, and A. Samsudin, "Compression-encryption based on fractal geometric," in *Proceedings of the 2nd International Conference on Computer Research and Development (ICCRD'10)*, pp. 213–217, May 2010.

[16] Y.-Y. Sun, R.-Q. Kong, X.-Y. Wang, and L.-C. Bi, "An image encryption algorithm utilizing Mandelbrot set," in *Proceedings of the 3rd International Workshop on Chaos-Fractal Theories and Applications (IWCFTA '10)*, pp. 170–173, IEEE, Yunnan, China, October 2010.

[17] J. B. Lima, E. A. O. Lima, and F. Madeiro, "Image encryption based on the finite field cosine transform," *Signal Processing: Image Communication*, vol. 28, no. 10, pp. 1537–1547, 2013.

[18] S. Rakesh, A. A. Kaller, B. C. Shadakshari, and B. Annappa, "Multilevel image encryption," https://arxiv.org/abs/1202.4871.

[19] M. M. de Souza, H. M. de Oliveira, R. M. de Souza, and M. M. Vasconcelos, "The discrete cosine transform over prime finite fields," in *Telecommunications and Networking—ICT 2004: 11th International Conference on Telecommunications, Fortaleza, Brazil, August 1–6, 2004. Proceedings*, vol. 3124 of *Lecture Notes in Computer Science*, pp. 482–487, Springer, Berlin, Germany, 2004.

[20] J. B. Lima, E. S. da Silva, and R. M. de Souza, "A finite field cosine transform-based image processing scheme for color image encryption," in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP '15)*, pp. 1071–1075, Orlando, Fla, USA, December 2015.

[21] B. Rama and J. Mishra, "Game-enabling the 3D-Mandelbulb fractal by adding velocity-induced support vectors," *International Journal of Computer Applications*, vol. 48, no. 1, pp. 1–3, 2012.

[22] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACIrandomness tests for image encryption," *Journal of Selected Areas in Telecommunications (JSAT)*, pp. 31–38, 2011.

[23] J. B. Lima, "Fast algorithm for computing cosine number transform," *Electronics Letters*, vol. 51, no. 20, pp. 1570–1572, 2015.

[24] The USC-SIPI Image Database, "University of Southern California, signal and image processing institute," http://sipi.usc.edu/database/.