I J C T A, 9(34) 2016, pp. 1-16

A Novel Image Encryption Based on Bit-Shuffled Improved Tent Map

Aaditya Gupta[*], Richa Thawait[*], K Abhimanyu Kumar Patro[**] and Bibhudendra Acharya[**]

ABSTRACT

With the fast progression of digital data communication in an electronic way, information security is becoming an issue in the modern advancing technology. Cryptography is one of the techniques of modern electronic security that serves the purpose of maintaining information security. There are numerous ways to encrypt and decrypt information for maintaining security. In this paper, a bit-shuffled improved tent map based image encryption technique is proposed. The implementation is based on the improved tent map combined with bit-shuffling operation using random sequence. The simulation results and security analysis shows that the proposed algorithm is resistive to brute-force attacks, statistical attacks, differential attacks, sensitive to secret keys and have more randomness. The proposed scheme is efficient and reliable and this technique can be applied in various fields where confidentiality of image is needed during communication and transmission such as in Governments, military, financial institutions, hospitals, etc.

Keywords: Image Encryption, Chaotic System, Improved Tent Map (ITM), Bit-Shuffling.

1. INTRODUCTION

Cryptography is the science of protecting the confidentiality of information during communication [1]. During the past years, a lot of image encryption techniques are used for protecting the confidentiality of image. The traditional image encryption techniques (for example, DES, IDEA, AES, etc.) are not appropriate for practical image encryption, especially for online communication because of bulkiness data capacity and strong correlation of pixels in the images [2, 3] and also a low-level of efficiency is achieved when the image is large [4]. Thus, there is need of new image encryption techniques. The chaos-based image encryption techniques are the current research hotspot in cryptography. Chaos systems have many important features such as high sensitivity to initial conditions, highly complex behavior, ergodicity, non-periodicity and determinacy [5]. Many numbers of chaotic image encryption techniques based on chaotic logistic map [6], chaotic standard map [7], chaotic perceptron model [8], 3D chaotic map [9] have been proposed. Fridrich [10] developed a permutation - substitution based image encryption model by using two-dimensional chaotic maps. Now days many of the image encryption systems are modeled by using this permutation-substitution principle. Permutation means changing the pixel positions in the original image. The changes occur either by using chaotically generated sequences or by using some matrix transformation methods such as Magic square transform, Arnold transform, etc [11]. These shuffling algorithms [12-17] successfully shuffles the positions of the pixels resulting better encryption but in some images only shuffling the pixel positions without changing of pixel values, leads to duplicacy in the histogram of encryption and the original image and thus its security could be threatened by the statistical analysis [2]. Substitution means changing the values of the pixels in the image. Only with the substitution [18-26], the encryption is not so good but in comparison with the permutation, the substitution may lead to higher security. Therefore, researchers combined both the permutation and substitution techniques to improve the encryption as well as the security

[*] Department of E & TC, NIT Raipur, Chhattisgarh, India, *E-mails: aadityagupta.nitrr@gmail.com; richathawait.29@gmail.com*

[**] Department of ETC, NIT Raipur, Chhattisgarh, India, *E-mails: abhimanyu.patro@gmail.com; bacharya.etc@nitrr.ac.in*

[2, 11]. In this paper, a bit-shuffled Improved Tent Map (ITM) based image encryption system has been modeled which is an extension of ITM [24]. The proposed scheme has one step permutation process and two step diffusion process architecture. Subsequent to the introduction in Section 1, Section 2 introduces some basic theories related to the proposed algorithm. The proposed methodology is presented in Section 3. Section 4 presents the simulation results and the security of the proposed methodology is analyzed. Finally, Section 5 concludes the paper.

2. PRELIMINARIES

In this section, first the TM, secondly the ITM, and finally the bit-shuffling operation are outlined.

2.1. The Tent Map

The Tent map also called as Triangle map which is a simple one-dimensional chaotic map [24] defined as

$$x_{n+1} = \begin{cases} 2x_n, \ if \ 0 \le x_n \le 1/2 \\ 2(1-x_n), if \ \ 1/2 \le x_n \le 1 \end{cases} \tag{1}$$

Due to simple and linear characteristic, it can be easily analyzed than the logistic map. But for certain values of the parameter, the map can give up complex chaotic behavior [27, 28]. Also the system is in a chaotic state when $x_0 \in (0, 1)$ [24].

So to improve the chaotic performance with in the interval (0, 1), the TM is generalized into an ITM [24].

2.2. The Improved Tent Map

The ITM [24] can be defined as

$$x_{n+1} = \begin{cases} (x_n - a \times floor(x_n / a)) / a, if \ \ floor(x_n / a) \, is \, even \\ (a \times (floor(x_n / a) + 1) - x_n) / a, \ if \ \ floor(x_n / a) \, is \, odd \end{cases} \tag{2}$$

where $x_n \in (0,1)$ $a \in (0, 0.5)$, a is the system parameter.

As highlighted by Liao [24], to study the dynamical behavior of the two systems such as system (1) and (2), a graph is plotted between x_n and the number of iterations, n at $x_0 = 0.23$ and $a = 0.24$. From that graph, it can be seen that, after 56 times of iteration, the state values of the system (1) are fall into $x_n = 0$. However, in system (2), the state values are distributed throughout the range (0, 1) randomly.

Similarly, Liao [24] argues that, by using Lyapunov exponent, one can find out the chaotic performance between the two systems such as system (1) and (2). A larger positive Lyapunov exponent results better chaotic performance. The system (2) has larger Lyapunov exponent than the system (1) resulting better chaotic performance of system (2) than the system (1).

Again, as highlighted by Liao [24], to measure the complexity of the two systems such as system (1) and system (2), C_0 complexity can be used. A system will be more complex if it has larger C_0 complexity value. From the complexity calculation [24] it can be seen that, the ITM system has larger complexity value than the TM system in a wider parameter range results more complexity value of system (2) than system (1).

From the above results it indicates that ITM system has excellent chaotic performance than the TM system.

2.3. Bit- Shuffling Operation

Bit-shuffling is a diffusion method where the values of the pixels are changed by shuffling the bits of each pixel. A pixel of gray scale image ranging from 0 – 256 is represented in 8-bit binary numbers. A random sequence of 8 numbers is generated and by using this sequence the bit positions are shuffled accordingly. Bits are shuffled according to the random sequence, so that more security is maintained and even a single bit change in the key will result in completely different image.

3. PROPOSED METHODOLOGY

This section presents the step-by-step procedure of bit-shuffled ITM method for image encryption and decryption. The proposed image encryption method poses three parts. In the first part, the permutation process based on ITM is performed as proposed by Liao [24]. In the second part, the bit-shuffling operation using ITM based random sequence is performed. Finally, in the last part, the diffusion process based on ITM is performed as proposed by Liao [24]. Similarly, the process for decryption also includes three parts. In the first part, the reverse diffusion process based on ITM is used as proposed by Liao [24]. In the second part, the bit-shuffling operation using ITM based random sequence is performed. Finally, in the last part, the reverse permutation process based on ITM is used as proposed by Liao [24]. The proposed image encryption and decryption process is as shown in Figure 1.

The plain gray-scale image having $L = M \times N$ size is used to analyze the results and security in the process of encryption. Let its data are represented into a one-dimensional vector $P = \{p(1), p(2), p(3) \cdots p(L-1), p(L)\}$. After permutation, let the image pixel sequence is represented as $P' = \{p'(1), p'(2), p'(3), \cdots, p'(L-1), p'(L)\}$ then, after bit-shuffling operation, let the image pixel sequence is represented as $P'' = \{p''(1), p''(2), p''(3), \ldots, p''(L-1), p''(L)\}$, and finally, let the ciphered-image pixel sequence is represented as $C = \{c(1), c(2), c(3), \ldots c(L-1), c(L)\}$. The secret key used in this scheme has seven parameters (x_{10}, a_1, x_{20}, a_2, x_{30}, a_3, s). The parameter is in relation with the plain-image. Furthermore, the two integers C_0 ($C_0 \in [1, 255]$) and N_0 ($N_0 > 500$) are also used.

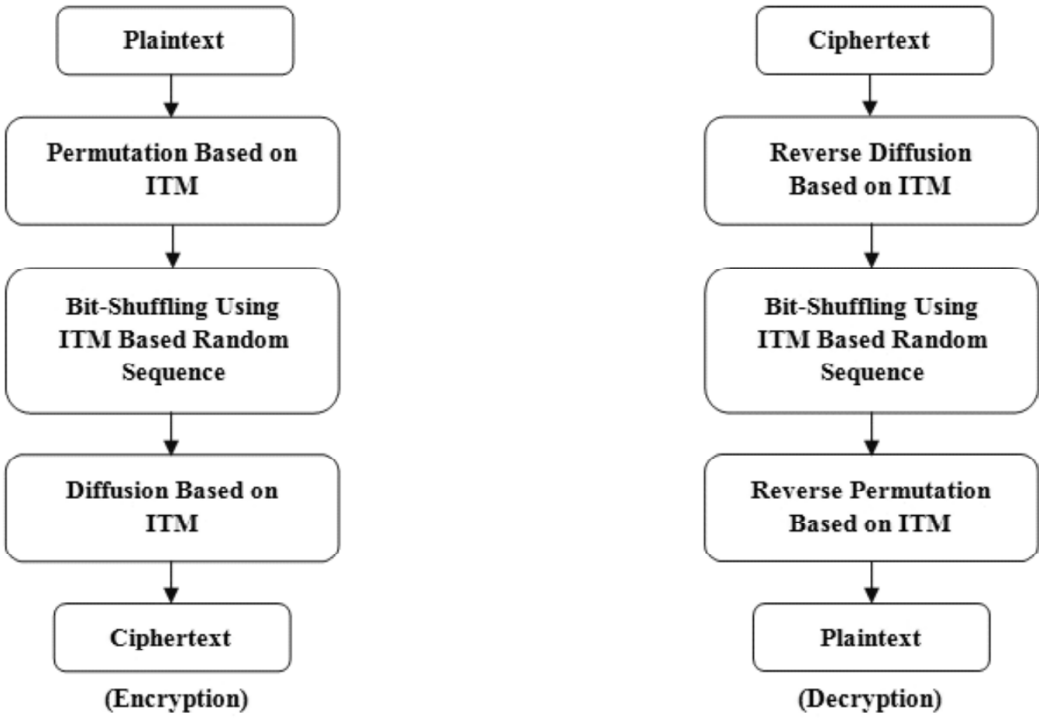

Figure 1: Bit-shuffled ITM based image encryption and decryption process

3.1. Encryption Algorithm

3.1.1. Permutation Process

The permutation process of the proposed encryption algorithm is same as the Liao's [24] designed ITM based permutation process which is described in section 3.1.1 of Reference [24]. In this part, initially, plain-image *P* and the parameters (x_{10}, a_1, *s*) are taken for the purpose of permutation process. In this part, at first, a sequence of integers $T = \{1(1), t(2), t(3), ..., t(L-1), t(L)\}$ are generated to permute the pixels in the plain-image *P*. These sequences of integers must satisfy the condition t(i) ¹ t(j) if $i \neq j$. *if*. The steps for permutation process are same as Reference [24], the only changes are as follows:

- In step 2 of section 3.1.1 of Reference [24], assign the value of *L* to *I* that means $I \leftarrow$ L. Therefore the ranges of *i* is from 1 to *I*.
- Step 11 is added where we assign back the value of to that means $L \leftarrow 1$.

3.1.2. Bit-Shuffling Process

The output of permutation process is used as the input of bit-shuffling process. Given the input matrix $P' = \{p'(1), p'(2), p'(3), \cdots, p'(L-1), p'(L)\}$, where $p'(i) \in [0, 255]$ and the paramerters (x_{20}, a_2). 8-bit binary of the pixel is represented by $Pbit = \{pbit(1), pbit(2), \ldots pbit(8)\}$, $pbit(i) \in [0, 1]$. The generated random sequence is denoted by $Rand = \{rand(1), rand(2), .. rand(8)\}$, where $rand(i) \in [1,8]$ and $rand(i) \neq rand(j)$ if $i \neq j$. Shuffled binary sequence is represented by $P'bit = \{p'bit(1), p'bit(2), \ldots p'bit(8)\}$, $p'bit(i) \in [0,1]$. *P dec* represents the decimal equivalent of 8-bit binary *P′bit*. The final output of bit-shuffling operation is represented by $P'' = \{p''(1), p''(2), p''(3), \ldots, p''(L-1), p''(L)\}$. The process for bit-shuffling operation is shown in Figure 2.

Step 1: $I \leftarrow 8, x \leftarrow x_{20} \times \frac{s_1}{I \times 255}$, where s_1 is the summation of 1 to 8, $a \leftarrow a_2$, $i \leftarrow 1$.

Step 2: Initialize $p'bit(j) \leftarrow 0, rand(j) \leftarrow j, j = 1,2, \cdots, 8.$.

Step 3: Obtain random sequence Rand by performing Step 3 to Step 10 of permutation algorithm stated in section 3.1.1 of Reference [24] using x, a only replacing T by Rand.

Step 4: By using the final generated value of *x* in step 3, iterating , 75 times to update *x*.

Step 5: Find by converting $p'(i)$ into 8-bit binary number.

Step 6: Shuffle *P bit* according to Rand generated in Step 3 using the following formula:

$$p'bit(j) = pbit\ (r\ and(j)),\ j = 1, 2, ..., 8. \tag{3}$$

Step 7: Find *P dec* by converting *P′bit* into its decimal equivalent.

Step 8: Assign $p''(i)$ as *Pdec*.

Step 9: Now increment the value of *i* by 1 that means $i \leftarrow$ i + 1, Repeat Step 2 to Step 9 until *i* reaches *L*.

P'' is the output of the bit-shuffling operation.

3.1.3. Diffusion Process

The diffusion process of the proposed encryption algorithm is same as the Liao's designed ITM based diffusion process and it is described in section 3.1.3 of Reference [24]. The technique used for diffusion process is bit-XOR. In this part, given the input matrix $P'' = \{p''(1), p''(2), p''(3), \cdots, p''(L-1), p''(L)\}$ and the parameters (x_{30}, a_3) are taken for the purpose of diffusion process. To change the pixel values in the input image P'', we generate different chaotic sequences by using (x_{30}, a_3). The diffusion process presented by

Liao [24] is of two rounds. The key sequence $K = \{k(1), k(2), k(3), \cdots, k(L-1), k(L)\}$ is generated in each round. In this diffusion process, cipher matrix generated as

$C = \{c(1), c(2), c(3), \ldots, c(L-1), c(L)$ from $P'' = \{p''(1), p''(2), p''(3), \cdots, p''(L-1), p''(L)\}$. The steps for diffusion process are same as Reference [24], the only changes are as follows:

- In step 1 of section 3.1.2 of Reference [24], we use x_{30} in place of x_{20}, a_3 in place of a_2 and $p''(i)$ in place of p'(i).

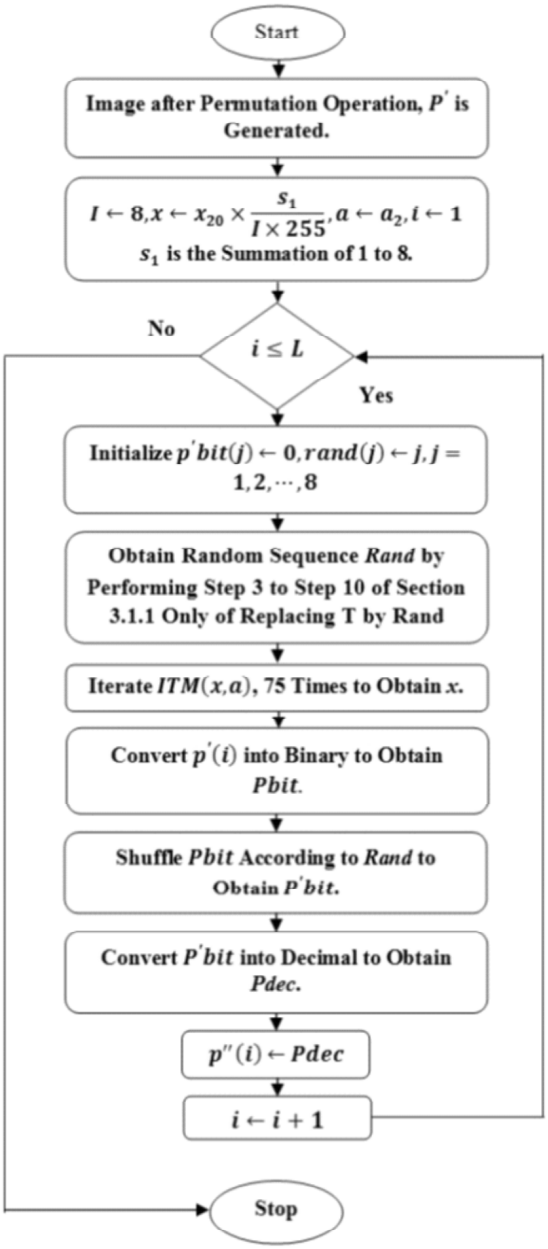

Figure 2: Process for bit-shuffling

3.2. Decryption Algorithm

The process for decryption is reverse process of encryption but the only modifications required in decryption are:

- The permutation process and diffusion process are executed in reverse order.

- In the diffusion process, the execution round is in reverse order that means $I = 2 \rightarrow 1$, and the pixel order is also in reverse that means $i = L \rightarrow 1$. The secret key parameters (x_{10}, a_1, x_{20}, a_2, x_{30}, a_3, s) and constants C_0 and N_0 are known.

3.2.1. Removing Diffusion Effect

The removing diffusion effect of the proposed algorithm is same as the Liao's designed removing diffusion effect and it is described in section 3.2 (1) of Reference [24]. In this process, we obtain the matrix $P'' = \{p''(1), p''(2), p''(3), \dots, p''(L-1), p''(L)\}$ from the cipher matrix $C = \{c(1), c(2), c(3), \dots, c(L-1), c(L)\}$. The steps for removing diffusion effect are same as Reference [24], the only changes are as follows:

- We use x_{30} in place of x_{20}, a_3 in place of a_2, $p''(i)$, in place of $p'(i)$ and P'' in place of P'.

3.2.2. Removing Bit-Shuffling Effect

In this process we obtain the matrix from $P' = \{p'(1), p'(2), p'(3), \cdots, p'(L-1), p'(L)\}$ from $P'' = \{p''(1), p''(2), p''(3), \dots, p''(L-1), p''(L)\}$.

Step 1: $I \leftarrow 8, x \leftarrow x_{20} \times \frac{s_1}{I \times 255}$, where s1 is the summation of 1 to 8, $a \leftarrow a_2$, $i \leftarrow 1$.

Step 2: Repeat Step 2 to Step 4 of bit-shuffling encryption algorithm described in sub-section 3.1.2.

Step 3: Find *Pbit* by converting $p''(i)$ into 8-bit binary.

Step 4: Shuffle according to *rand* generated in Step 2 by using the following formula:

$$p'bit\left(rand(j)\right) = pbit(j), j = 1,2, \cdots, 8. \tag{4}$$

Step 5: Find *Pdec* by converting *P'bit* into decimal.

Step 6: Assign as p'(i) as *Pdec*.

Step 7: $i \leftarrow i + 1$, repeat Step 2 to Step 6 until reaches *L*.

3.2.3. Removing Permutation Effect

In this process, we obtain the matrix $P = \{p(1), p(2), p(3), \dots, p(L-1), p(L)\}$ from the matrix $P' = \{p'(1), p'(2), p'(3), \dots, p'(L-1), p'(L)\}$. All operations are the same as Step 2 to Step 11 in the permutation process except that equation in Step 4 is replaced by

$$p(1) \leftarrow p'(j), p(j) \leftarrow p'(1), t(1) \leftarrow j, t(j) \leftarrow 1. \tag{5}$$

and equation in Step 9 is replaced by:

$$p(j) \leftarrow p'(i), p(i) \leftarrow p'(j); \tag{6}$$

$$temp \leftarrow t(i), t(i) \leftarrow t(j), t(j) \leftarrow temp. \tag{7}$$

4. SIMULATION RESULTS AND SECURITY ANALYSIS

In this paper, the two standard gray level images 'Cameraman' and 'Lena' having dimensions 256×256 are used. The simulations are carried out by using MATLAB R2012a on a PC with an Intel CORE i5, 1.60 GHz CPU, 4.00 GB memory and 500 GB hard disk with windows 8 operating system and the secret key parameters are taken as $x_{10} = 0.3527$, $x_{20} = 0.34273$, $x_{30} = 0.7278562889573$, $a_1 = 0.774$, $a_2 = 0.65674$, $a_3 = 0.676$ and s. The constant parameters are taken as $N_0 = 1000$, $C_0 = 211$. The simulation results are as shown in Figure 3. In Figure 3 (b) and (e), it found that the proposed algorithm can able to encrypt the images properly and from Figure 3 (c) and (f), it found that the proposed algorithm can also able to decrypt the images properly. From the visual point of view, there is no relationship between original images and their corresponding encrypted images. It is obvious that our algorithm provides good encryption effect.

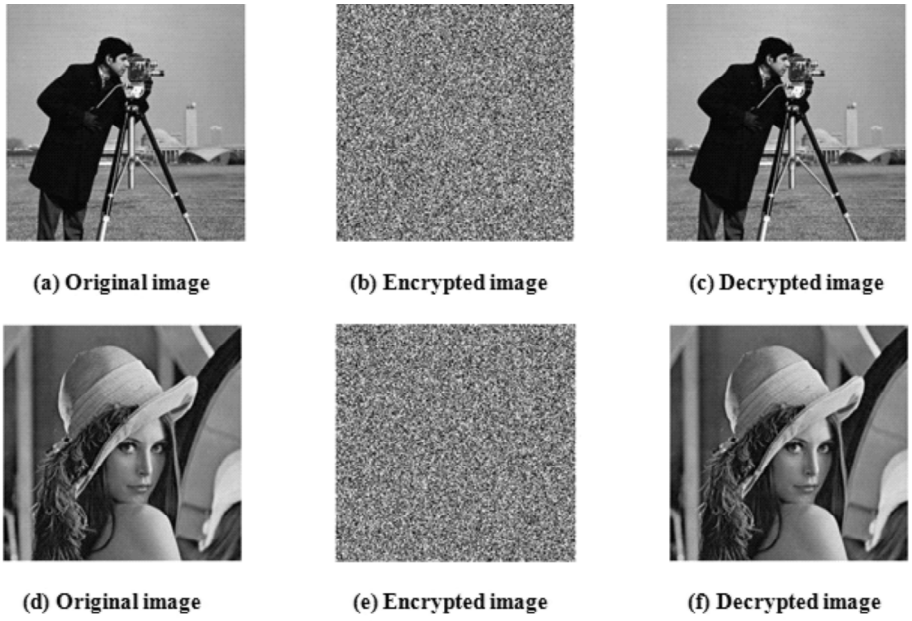

Figure 3. (a), (b), and (c) Simulation results of 'Cameraman' image, (d), (e), and (f) Simulation results of 'Lena' image by using the proposed method

Security is the main purpose of all encryption algorithms. A good encryption algorithm should be designed to resist all kinds of known attacks, such as exhaustive attack, statistical attack and differential attack. Basically, the most important analysis is the key space and key sensitivity. Key space should be large to resist brute-force attacks and key sensitivity should be high. This section briefly analyzes the security of the proposed image encryption algorithm.

4.1. Key Space Analysis

Key space is the total number of different keys used in the encryption process [29]. It should be large enough ($> 2^{128}$) to resist brute-force attack [29, 30]. A large key space effectively resists the algorithm from exhaustive attack [31]. In the proposed algorithm, the secret keys include $\{x_{10}, a_1, x_{20}, a_2, x_{30}, a_3, s\}$. They all are double–precision numbers except s. If the precision is 10^{16} then the total key space will be $10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} = 10^{96}$, which is large enough to prevent exhaustive searching and also much larger than the total key space generated by Liao [24]. Thus, brute-force attack is computationally difficult.

4.2. Statistical Analysis

In order to resist the statistical attacks, the encrypted images should possess certain random properties.

4.2.1. Histogram Uniformity of Encrypted Image

Histogram is a graph between the pixel intensities and the number of pixels in an image. The statistical attack is very effective if evenness or uniformity occurs in a histogram of an image [32]. It is desirable that after encryption the pixel grey values are to be scatter in the entire pixel value space. The histograms of 'Cameraman' and 'Lena' are shown in Figure 4 and 5 respectively by using the proposed method. We compare the gray histogram of the image before and after encryption to analyze the statistical performance. Figure 4(a) and 5(a) represents the gray histograms of original images and Figure 4(b) and 5(b) represents the gray histograms of corresponding encrypted images. From the two set of figures, we can see that the original pixel gray values are concentrated on some value, but the pixel gray values after the encryption are scattering in the entire pixel value space, namely, two images have lower similarity. As a result the image is

not too dark and not too bright and offers good contrast. Hence, it is difficult to use the statistical performance of the pixel gray value to recover the original image. Thereby, the proposed method has strong ability to resist statistical attack. From the histograms of Figure 4(c) and 5 (c), it is clear that there is no loss of data in encryption and decryption process. Hence the proposed method is applicable to protect conventional images during communication and transmission.

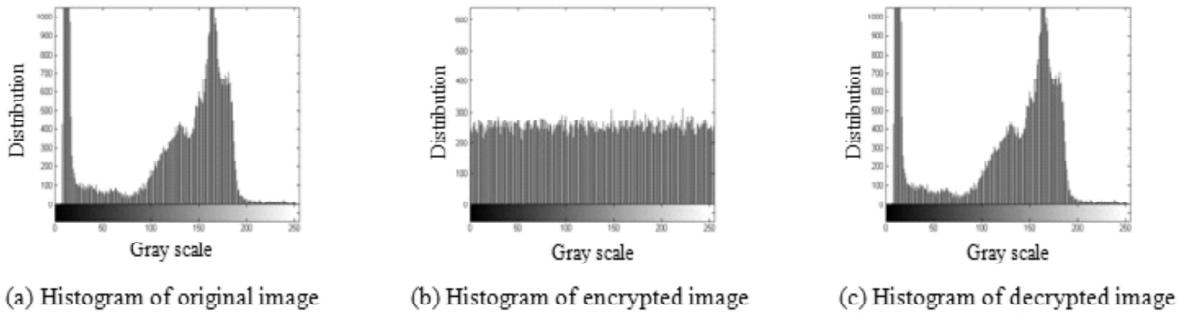

Figure 4: Histograms of 'Cameraman' image by using the proposed method

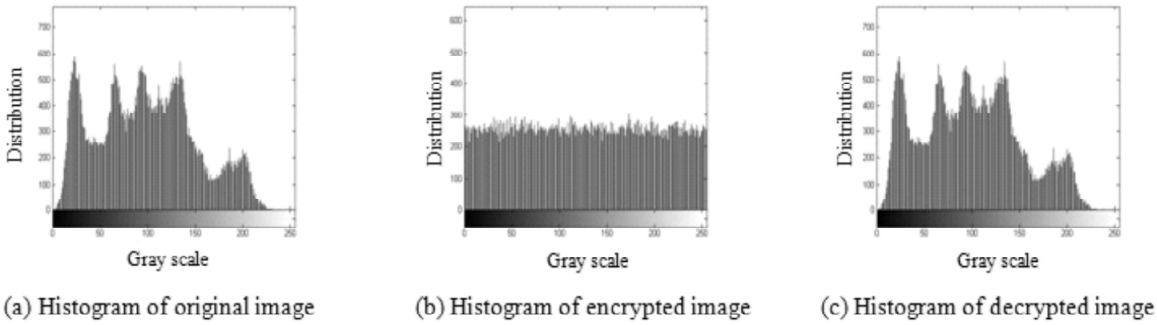

Figure 5: Histograms of 'Lena' image by using the proposed method

4.2.2. Histogram Deviation (HD) Between Original Image and Encrypted Image

It measures the amount of deviation occurs between the original images and the encrypted images. Higher value of *HD* indicates higher encryption accuracy [36]. Table 1 shows the higher value of *HD* which indicates higher encryption accuracy. However, this measure is not sufficient to use for measuring encryption quality because it only measures the difference between the histograms of the original and encrypted images [36].

The mathematical expression to measure *HD* is as given below.

$$HD = \frac{\left(\frac{d_0 + d_{255}}{2} + \Sigma_{i=1}^{255} d_i \right)}{M \times N} \tag{8}$$

where d_i is the amplitude of the absolute difference at the i^{th} grey level.

4.2.3. Irregular Deviation (ID) Between Original Image and Encrypted Image

It measures the quality of encryption in terms of how much the deviation caused by encryption (on the encrypted image) is irregular. The difference image is calculated by calculating the absolute difference between the original and encrypted image. Lower value of *ID* indicates higher encryption accuracy [36]. Table 1 shows the lower value of *ID* which indicates higher encryption accuracy.

The mathematical expression to measure *ID* is as described below.

$$ID = \frac{\Sigma_{i=0}^{255} | H(i) - M_H |}{M \times N} \tag{9}$$

where M_H is the mean value of histogram and

H is the histogram of the difference image.

4.2.4. Deviation from Identity (DI) Between Original Image and Encrypted Image

The histogram of an ideally encrypted image should have uniformly distribution throughout the grey levels. The *DI* metric measures the deviation of the histogram of the encrypted image from the histogram of an image, which is ideally encrypted. Lower value of *DI* indicates better encryption quality [36]. Table 1 shows the lower value of *DI* which indicates higher encryption accuracy.

The mathematical expression to measure *ID* is as given below.

$$DI = \frac{\Sigma_{i=0}^{255} | H(C_I) - H(C) |}{M \times N} \tag{10}$$

where $H(C)$ is the histogram of encrypted image and

$$H(C_I) = \begin{cases} \dfrac{M \times N}{256} & , 0 \le C_I \le 255 \\ 0, otherwise \end{cases} \tag{11}$$

Table 1
HD, ID, and DI criteria of the proposed bit-shuffled ITM method

Criteria (expected value)	*Image*	*Proposed bit-shuffled ITM method*
HD (Higher value)	Cameraman	2.5000
	Lena	2.5000
ID (Lower value)	Cameraman	1.9844
	Lena	1.9844
DI (Lower value)	Cameraman	0.9922
	Lena	0.9919

4.2.5. Scattered Diagram Between Original Vs. Encrypted, Original Vs. Decrypted Images

Figure 6(a) and (c) shows the scattered diagram between original and encrypted images by using the proposed method. From the figure it is clear that, the points are not in a line, it spreads throughout the surface. That means weaker correlation occurs between original and encrypted images. Figure 6(b) and 6(d) shows the scattered diagram between original and decrypted images. From the figure it is clear that, all the points are along a line. That means stronger correlation occurs between original and decrypted images.

4.2.6. Correlation of Adjacent Pixels

Correlation coefficient finds the degree of linear correlation between two adjacent pixels [33]. Simply, it reflects the degree of image scrambling [34]. The range of the correlation coefficient r is [–1, 1]. If the value of r greater than zero then it indicates positive correlation and if the value of r is less than zero then it indicates negative correlation. $|r|$ which represents the degree of correlation between two adjacent pixels,

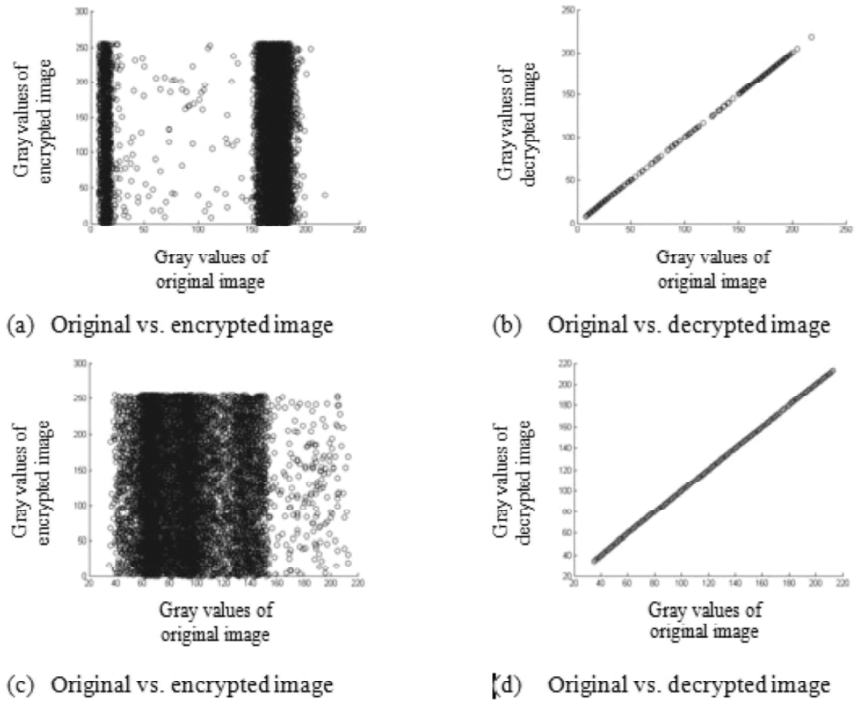

Figure 6: (a) and (b) Scattered diagram of 'Cameraman' image, (c) and (d) Scattered diagram of 'Lena' image by using the proposed method

with indicating perfect correlation and $r = 0$ indicating uncorrelated pixels [33]. The correlation between the two adjacent pixels in an original image is almost close to 1. An effective encryption algorithm should reduce the correlation between adjacent pixels which is almost close to 0 no matter in horizontal, vertical and diagonal directions [34]. That means the adjacent pixels of original image have very strong linear correlation, while the correlation between adjacent pixels of encrypted image is very small.

The mathematical expressions to calculate the correlation coefficient of two adjacent pixels is

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{12}$$

where

$$\text{cov}(x, y) = \frac{1}{N}\Sigma_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \tag{13}$$

$$E(x) = \frac{1}{N}\Sigma_{i=1}^{N}x_i, \tag{14}$$

$$D(x) = \frac{1}{N}\Sigma_{i=1}^{N}(x_i - E(x))^2 \tag{15}$$

About 3000 thousand pairs of two adjacent (in vertical, horizontal, and diagonal direction) pixels are randomly selected from the encrypted image, and the correlation coefficients are calculated, which are as shown in Table 2. From Table 2 we concluded that the correlation coefficient of original image is almost close to 1 and the correlation coefficient of encrypted images by using the proposed method is close to 0 in all the three directions that means no similarity occurs between original images and encrypted images. Finally from the tabulated data, we concluded that adjacent pixels of original image have very strong linear correlation, while the correlation between adjacent pixels of encrypted image using the proposed method is

very small. It has damaged the linear correlation of original image. Therefore the encrypted algorithm can effectively resist pixel correlation statistical attack. Figure 7 and 8 shows the correlation distribution of two adjacent pixels for 'Cameraman', and 'Lena' images respectively. From the contrast diagrams we can observe that the correlation between pixels of original image is much larger than the correlation between pixels of encryption image. That means, the adjacent pixels of original image have very strong linear correlation, while the correlation between adjacent pixels of encrypted image is very small. It has damaged the linear correlation of original image. Therefore the proposed encrypted method can effectively resist pixel correlation statistical attack.

Table 2
Comparison of correlation coefficients between ITM method and the proposed bit-shuffled ITM method

Correlation coefficient	*Plain images*		*Encrypted images by using ITM method [24]*		*Encrypted images by using proposed bit-shuffled ITM method*	
	Cameraman	*Lena*	*Cameraman*	*Lena*	*Cameraman*	*Lena*
Horizontal (H)	0.9335	0.9400	-	0.000272	0.0017	-0.0043
Vertical (V)	0.9592	0.9693	-	0.000735	-0.0162	0.0080
Diagonal (D)	0.9087	0.9179	-	0.002389	-0.0014	0.0018
$(H2 + V2 + D2)^{0.5}$	1.6178	1.6327	-	0.002514	0.0164	0.0092
Average (H, V, D)	0.9338	0.9424	-	0.001132	-0.0053	0.0018

4.3. Differential Analysis

The major requirement of all the encryption techniques is that the encrypted image should be significantly different from the original one. To quantify the difference between encrypted image and the corresponding original image, three measures were used: Mean Absolute Error (MAE), the Number of Pixel Change Rate (NPCR), and Unified Average Changing Intensity (UACI).

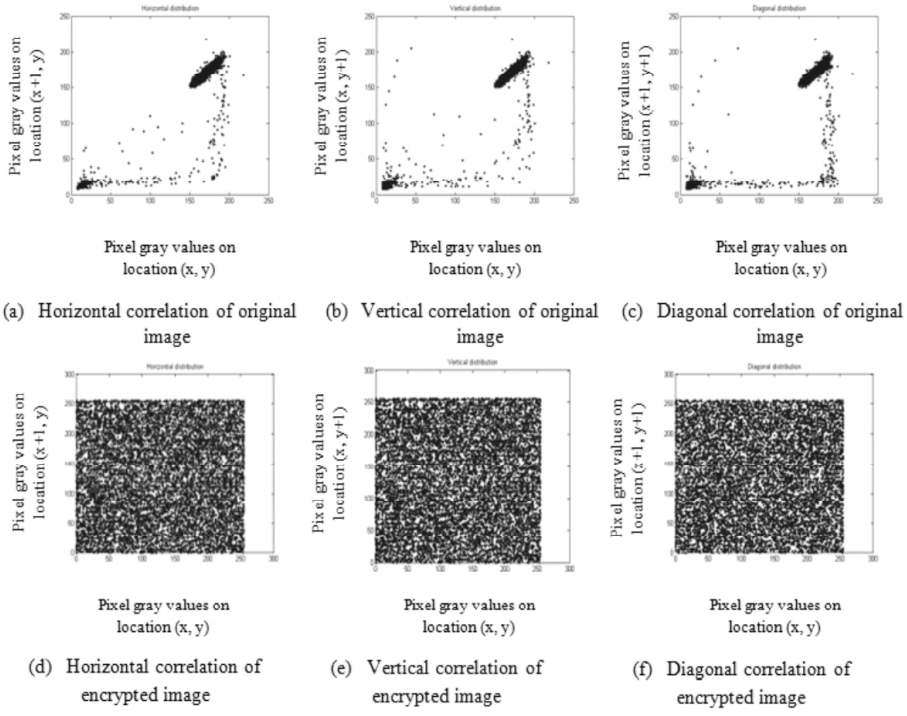

Figure 7: (a), (b), and (c) Correlation distribution of two adjacent pixels for original 'Cameraman' image, (d), (e), and (f) Correlation distribution of two adjacent pixels for encrypted 'Cameraman' image

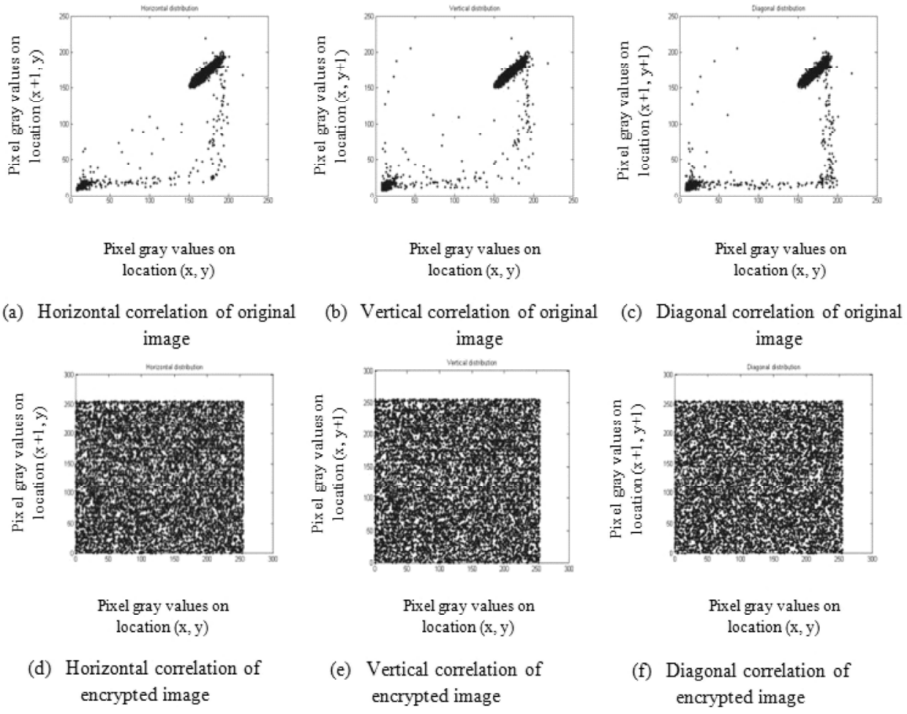

Figure 8: (a), (b), and (c) Correlation distribution of two adjacent pixels for original 'Lena' image, (d), (e), and (f) Correlation distribution of two adjacent pixels for encrypted 'Lena' image.

The MAE between plain image and encrypted image is given by the relation

$$MAE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} |a_{ij} - b_{ij}| \tag{16}$$

where $M \times N$ is the size of the original and encrypted images. The parameters a_{ij} and b_{ij} are gray scale values of pixels in original and encrypted images, respectively. The larger the MAE value, the better is the encryption security.

NPCR is the change rate of the encrypted image pixels when the image changes one pixel in the process of encryption. The more NPCR gets close to 100%, the more sensitive the cryptosystem to the changing of plain image and the more effective for the cryptosystem to resist plaintext attack [35].

The mathematical expression to calculate NPCR is

$$\mathrm{NPCR} = \frac{1}{\mathrm{W} \times \mathrm{H}} \sum_{\mathrm{i}=1}^{\mathrm{W}} \sum_{\mathrm{j}=1}^{\mathrm{H}} \mathrm{D}(\mathrm{i},\mathrm{j}) \times 100\% \tag{17}$$

where,

$$\mathrm{D}(\mathrm{i},\mathrm{j}) = \begin{cases} 0, \text{if } \mathrm{C}_1(\mathrm{i},\mathrm{j}) = \mathrm{C}_2(\mathrm{i},\mathrm{j}) \\ 1, \text{if } \mathrm{C}_1(\mathrm{i},\mathrm{j}) \neq \mathrm{C}_2(\mathrm{i},\mathrm{j}) \end{cases}, \tag{18}$$

W and H represents the width and height of the images respectively, C_1 and C_2 are the respective cipher images before and after one pixel changed in a plain image for the pixel at position (i,j). For a 256 gray-scale image, the expected that means the ideal value of NPCR is found to be 99.6094%. The larger is the value; the better is the encryption quality.

UACI is the change rate of the average strength of the original image and the encrypted image. The larger UACI is the more effective for the cryptosystem to resist differential attack [35].

The mathematical expression to calculate UACI is

$$\mathrm{UACI} = \frac{1}{\mathrm{W} \times \mathrm{H}} \left(\sum_{\mathrm{i}=1}^{\mathrm{W}} \sum_{\mathrm{j}=1}^{\mathrm{H}} \frac{|\mathrm{C}_1(\mathrm{i},\mathrm{j}) - \mathrm{C}_2(\mathrm{i},\mathrm{j})|}{255} \right) \times 100\% \tag{19}$$

Where W and H represents the width and height of the images respectively, C_1 and C_2 are the respective cipher images before and after one pixel changed in a plain image for the pixel at position (i, j). For a 256 gray-scale image, the expected that means the ideal value of UACI is found to be 33.4635 %. The larger is the value; the better is the encryption quality.

Liao [24] reported the mean NPCR and UACI values are 99.6062% and 33.3970%, respectively. In the proposed scheme, the mean NPCR of 'Cameraman' and 'Lena' are 99.615173% and 99.616160% respectively and the mean UACI of 'Cameraman' and 'Lena' are 33.4953% and 33.4818% respectively which is larger than the value of NPCR and UACI of Liao [24]. The NPCR, UACI and MAE of 'Cameraman' and 'Lena' images by using the proposed bit-shuffled ITM algorithm are tabulated in Table 3.

Table 3
NPCR, UACI, and MAE criteria of the proposed bit-shuffled ITM method

Criteria (expected value)	*Images*	*Proposed bit-shuffled ITM method*
NPCR (99.6094%) (Mean value)	Cameraman	99.615173
	Lena	99.616160
UACI (33.4635%) (Mean value)	Cameraman	33.4953
	Lena	33.4818
MAE (Larger Value)	Cameraman	79.4729
	Lena	77.6965

4.4. Key Sensitivity Analysis

4.4.1. Sensitivity to the Plaintext

To test the sensitivity to the plaintext, we randomly choose one pixel of the plain image 'Cameraman' and 'Lena' and then calculate NPCR and UACI between each pair of cipher images. We are taken 100 pairs of cipher images and the NPCR and UACI results of 'Cameraman' and 'Lena' are shown in Figure 9 and Figure 10 respectively. The mean value of NPCR for 'Cameraman' and 'Lena' are 99.615173 and 99.616160

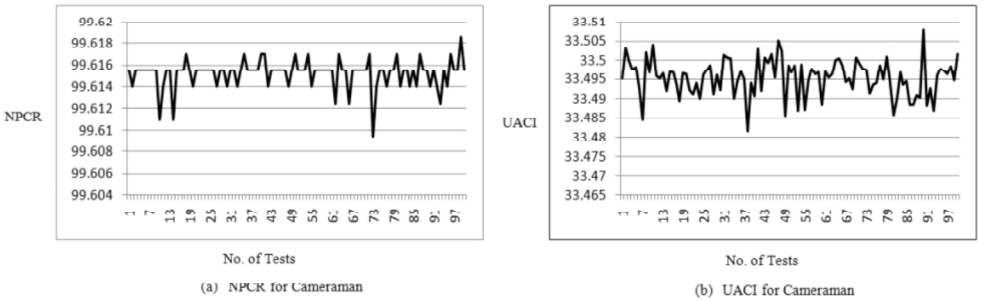

Figure 9: (a) and (b) NPCR and UACI graph chart for 'Cameraman' image respectively

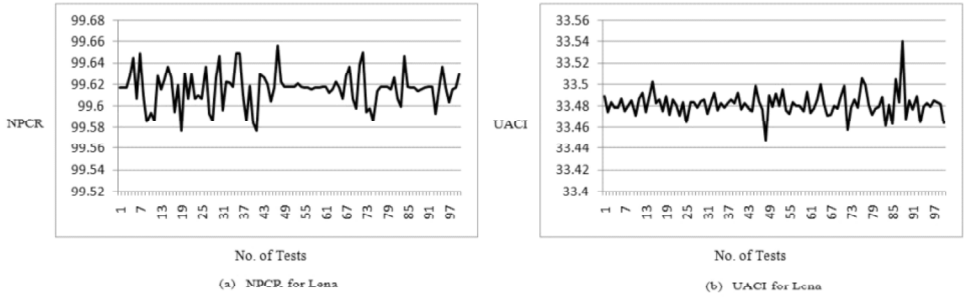

Figure 10: (a) and (b) NPCR and UACI graph chart for 'Lena' image respectively

respectively and the mean value of UACI for 'Cameraman' and 'Lena' are 33.4953 and 33.4818 respectively. It is clear that the NPCR and UACI values remain in the vicinity of the expected values, which means, our proposed encryption scheme is very much sensitive to the plaintext.

4.4.2. Sensitivity to the Secret Keys

To test the sensitivity to the secret keys, the NPCR and UACI between two encrypted images with keys $(x_{10}, a_1, x_{20}, a_2, x_{30}, a_3) = (0.3527, 0.774, 0.34273,\ 0.65674, 0.7278562889573, 0.676)$ and slightly varied keys (only one of the six parameters has varied 10^{-10}) are calculated and the results are as shown in Table 4. From Table 4 we can observe that the calculated NPCR and UACI are close to the ideal values. Therefore, the encryption algorithm is very sensitive to the secret keys.

Table 4
NPCR and UACI values with slightly varied keys

Modified Keys	$\Delta x_{10}=10^{-10}$	$\Delta a_1= 10^{-10}$	$\Delta x_{20}=10^{-10}$	$\Delta a_2= 10^{-10}$	$\Delta x_{30}= 10^{-10}$	$\Delta a_3= 10^{-10}$
NPCR	99.6613	99.6277	99.7070	99.6399	99.5712	99.5621
UACI	33.5213	33.5338	32.7412	33.6704	33.5228	33.6601

4.5. Information Entropy Analysis

Information entropy measures the randomness of an image which is used to characterize the texture of an image [32]. The mathematical expression for calculating information entropy is:

$$H(m) = \Sigma_{i=0}^{M-1}\ p(m_i) \log \frac{1}{p(m_i)} \tag{20}$$

Where m is the source of information, M is the total number of symbols $m_i \in m$ and $p(m_i)$ and denotes the probability of symbols [32]. If the information source sends 256 symbols then the theoretical value of entropy will be $H(m) = 8$ [32]. The closer it gets to 8, the harder for the attackers to decode cipher images. Table 5 presents the comparison of information entropy of original image and encrypted image by using ITM system [24] and the proposed bit-shuffled ITM system. A higher value of the entropy obtained in case of our proposed method as compared to that obtained in ITM system [24] indicates that our algorithm provides more randomness in the encrypted image resulting in better encryption.

Table 5
Information Entropy of original images and encrypted images by using ITM system [24] and the proposed bit-shuffled ITM system

Images	*Original images*	*Encrypted images*	
		ITM based system [24]	*Proposed bit-shuffled ITM system*
Cameraman	7.0097	-	7.9973
Lena	7.5691	7.9978	7.9981

5. CONCLUSIONS

In this paper, an extended ITM based image encryption technique is introduced. The proposed technique includes ITM based permutation, ITM based bit-shuffling operation, and then ITM based diffusion. Results of this technique have been analyzed and it has been observed that the proposed scheme possesses large key space than the ITM technique which proves that the proposed algorithm resists brute-force attack more effectively. From the key sensitivity analysis, it can be observed that, the proposed technique is more

sensitive to the plain-images and also to the secret keys. NPCR, UACI, and entropy values are also improved and are much better than the Liao's ITM method. So, the proposed image encryption scheme has strong ability of resisting all the known attacks. This shows that the proposed scheme is best and is more appropriate for image encryption.

ACKNOWLEDGMENTS

This research work is supported by Information Security Education Awareness (ISEA) project phase – II, Department of Electronics and Information Technology (DeitY), Govt. of India.

REFERENCES

[1] M. Prasad, and K. L. Sudha, "Chaos image encryption using pixel shuffling," Computer Science & Information Technology (CS & IT) CCSEA, pp. 169-179, 2011.

[2] Q. Zhang, L. Guo, and X. Wei, "Image encryption using DNA addition combining with chaotic maps," Mathematical and Computer Modelling, Vol. 52, no. 11, pp. 2028-2035, 2010.

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

[4] H. Gao, Y. Zhang, S. Liang, and D. Li, "A new chaotic algorithm for image encryption," Chaos, Solitons & Fractals, Vol. 29, no. 2, pp. 393-399, 2006.

[5] R. Guesmi, M. A. Farah, A. Kachouri, and M. Samet, "A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2," Nonlinear Dynamics, Vol. 83, no. 3, pp. 1123-1136, 2016.

[6] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," Image and Vision Computing, Vol. 24, no. 9, pp. 926-934, 2006.

[7] K. W. Wong, B. S. Kwok, and W. S. Law, "A fast image encryption scheme based on chaotic standard map," Physics Letters A, Vol. 372, no. 15, pp. 2645-2652, 2008.

[8] X. Y. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," Nonlinear Dynamics, Vol. 62, no. 3, pp. 615-621, 2010.

[9] A. Kanso, and M. Ghebleh, "A novel image encryption algorithm based on a 3D chaotic map," Communications in Nonlinear Science and Numerical Simulation, Vol. 17, no. 7, pp. 2943-2959, 2012.

[10] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," International Journal of Bifurcation and chaos, Vol. 8, no. 06, pp. 1259-1284, 1998.

[11] Q. Zhang, X. Xue, and X. Wei, "A novel image encryption algorithm based on DNA subsequence operation," The Scientific World Journal, 2012.

[12] X. Wang, and J. Zhang, "An image scrambling encryption using chaos-controlled Poker shuffle operation," International Symposium in Biometrics and Security Technologies 2008 (ISBAST 2008) IEEE, pp. 1-6, 2008.

[13] M. Li, T. Liang, and Y. J. He, "Arnold transform based image scrambling method," 3rd International Conference on Multimedia Technology, 2013.

[14] R. Huang, and K. Sakurai, "A robust and compression-combined digital image encryption method based on compressive sensing," Seventh International Conference in Intelligent Information Hiding and Multimedia Signal Processing 2011 (IIH-MSP 2011) IEEE, pp. 105-108, 2011.

[15] W. Hui-qin, H. Ji-chao, and C. Fu-ming, "Colour Image Watermarking Algorithm Based on the Arnold Transform," International Conference in Communications and Mobile Computing 2010 (CMC 2010), IEEE, Vol. 1, pp. 66-69, 2010.

[16] L. Wu, J. Zhang, W. Deng, D. He, "Arnold transformation algorithm and anti-Arnold transformation algorithm," 1st International Conference in Information Science and Engineering 2009 (ICISE 2009) IEEE, pp. 1164-1167, 2009.

[17] W. Hui-qin, H. Ji-chao, and C. Fu-ming, "Colour Image Watermarking Algorithm Based on the Arnold Transform," International Conference in Communications and Mobile Computing 2010 (CMC 2010), Vol. 1, pp. 66-69, 2010.

[18] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (Blowfish)," International Workshop on Fast Software Encryption, pp. 191-204, Springer Berlin Heidelberg, 1993.

[19] R. Anderson, M. Kuhn, "Low cost attacks on tamper resistant devices," In International Workshop on Security Protocols, pp. 125-136, Springer Berlin Heidelberg, 1997.

[20] Y. Wang, K. W. Wong, X. Liao, and T. Xiang, "A block cipher with dynamic S-boxes based on tent map," Communications in Nonlinear Science and Numerical Simulation, Vol. 14, no. 7, pp. 3089-3099, 2009.

[21] X. Yi, C. H. Tan, and C. K. Siew, "A new block cipher based on chaotic tent maps," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 49, no. 12, pp. 1826-1829, 2002.

[22] S. S. Askar, A. A. Karawia, and A. Alshamrani, "Image encryption algorithm based on chaotic economic model," Mathematical Problems in Engineering, 2015.

[23] H. Khanzadi, M. Eshghi, and S. E. Borujeni, "Image encryption using random bit sequence based on chaotic maps," Arabian Journal for Science and engineering, Vol. 39, no. 2, pp. 1039-47, 2014.

[24] X. Liao, "Improved Tent Map and Its Applications in Image Encryption," International Journal of Security and Its Applications, Vol. 9, no. 1, pp. 25-34, 2015. doi:10.14257/ijsia.2015.9.1.03.

[25] H. S. Kwok, and W. K. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," Chaos, solitons & fractals, Vol. 32, no. 4, pp. 1518-1529, 2009.

[26] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," Optics and Lasers in Engineering, Vol. 78, pp. 17-25, 2016.

[27] S. Lynch, Dynamical systems with applications using Mathematica®. Springer Science & Business Media, 2007.

[28] [Online] The World is Mysterious. *https://theworldismysterious.wordpress.com/2013/10/11/chaos-theory-tent-map-part-1/*.

[29] V. Patidar, N. K. Pareek, G. Purohit, and K. K. Sud, "A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption," Optics Communications, Vol. 284, no. 19, pp. 4331-4339, 2011.

[30] A. Kulsoom, D. Xiao, and S. A. Abbas, "An efficient and noise resistive selective image encryption scheme for gray images based on chaotic maps and DNA complementary rules," Multimedia Tools and Applications, Vol. 75, no. 1, pp. 1-23, 2016.

[31] X. Li, L. Wang, Y. Yan, and P. Liu, "An improvement color image encryption algorithm based on DNA operations and real and complex chaotic systems," Optik-International Journal for Light and Electron Optics, Vol. 127, no. 5, pp. 2558-2565, 2016.

[32] X. Wang, and H. L. Zhang, "A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems," Nonlinear Dynamics, Vol. 83, no. 1-2, pp. 333-346, 2016.

[33] J. Zhang, D. Fang, and H. Ren, "Image encryption algorithm based on DNA encoding and chaotic maps," Mathematical Problems in Engineering, 2014.

[34] Q. Zhang, and X. Wei, "RGB color image encryption method based on Lorenz chaotic system and DNA computation, IETE Technical Review, Vol. 30, no. 5, pp. 404-409, 2013.

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

[36] C. Chattopadhyay, B. Sarkar, and D. Mukherjee, "Encoding by DNA Relations and Randomization Through Chaotic Sequences for Image Encryption," arXiv preprint arXiv:1505.01795, 2015.