International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

A Nature Inspired Color Image Encryption Technique to Protect the Satellite Images

Bhagyashri Pandurangi R†* and Meenakshi R. Patil

†Department of Electronics and Communication Engineering, KLS Gogte Institute of Technology, Belagavi, Karnataka, India
‡Department of Electronics and Communication Engineering, Jain AGMIT, Jamakhandi, Karnataka, India

Received 01 March 2019, Accepted 01 May 2019, Available online 02 May 2019, **Vol.9, No.3 (May/June 2019)**

Abstract

A color image encryption algorithm based on chaotic maps is proposed in this paper. The algorithm is based on two bio-operations: crossover and mutation. To enhance the robustness against differential attacks, the mutated image is subjected to scrambling process operated on the pixel values of the image using a random sequence. Experimental results show that the proposed algorithm is capable of generating encrypted images with uniform distribution of the pixel values and very low correlation coefficients of adjacent pixels. It is very sensitive to any change in the secret key values. The results show that the algorithm is robust to statistical and differential attacks.

Keywords: *Cryptography, symmetric encryption, logistic map, bio inspired encryption, digital transactions*

1. Introduction

Due to phenomenal growth in internet, multimedia content such as image, video and audio can be easily transmitted from source to destination. This develops a huge impact on development of applications including e-commerce, industry, multimedia and entertainment which rely totally on the Internet. Thus, to stop corruption of multimedia transmission content from unauthorized user, security plays a major role for content protection. In the past few years, cryptography has become a major tool for securing multimedia transmission content such as image, video and audio. Image encipherment is somehow different from text encryption due to some typical image characteristics like bulkiness and high correlation among pixels, which are generally difficult to handle by traditional methods [Sahar Mazloom, 2011]. The properties of chaotic systems like sensitivity to initial conditions, pseudorandom nature, and non-periodicity, have selected them a suitable tool for image encryption [Z. Lin, 2009].

A region based selective image encryption technique was proposed in [K. C. Ravishankar,2006] that provides the facility of selective encryption and selective reconstruction of images. A modified chaotic based image encryption scheme using two logistic maps and two-dimensional baker map was proposed in [Rashidah Kadir, 2010]. Paper [Sahar Mazloom, 2011] proposes a symmetric image cipher based on the widely used confusion–diffusion architecture which utilizes the chaotic 2D Standard map and 1D Logistic map. Paper [Aihong Z, 2010] proposed a method of color image protective transmission based on Logistic map and LSB hiding algorithm. A new self-adaptive image encryption algorithm was presented in [Chen Gang, 2005] that takes on a thorough integrity protect function and can be used in data validation.

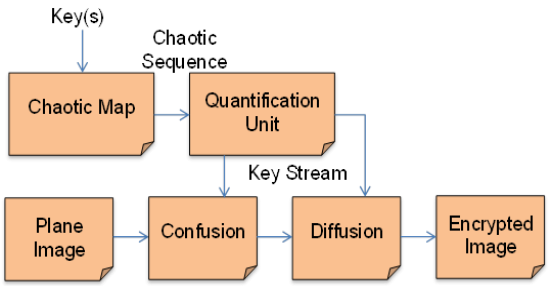

Fig.1 Typical architecture of an encryption algorithm

Based on self-adaptive wave transmission, an image encryption algorithm was given in the paper [Xiaofeng Liao, 2010]. In the paper [Shujiang Xu, 2010] fast image encryption scheme based on a nonlinear chaotic map was proposed. Liu and Wang have designed a stream-cipher algorithm based on one-time keys and robust chaotic maps, in order to get high security and improve the dynamical degradation [Liu Hongjun, 2010]. They have utilized the piecewise linear chaotic map as the generator of a pseudo-random key stream sequence. The various image encryption algorithms proposed in

*Corresponding author's ORCID ID: 0000-0001-9620-3571
DOI: https://doi.org/10.14741/ijcet/v.9.3.4

different papers vary in the type of chaotic maps used, confusion and diffusion methods, key size, assuming 2D or 3D image representation, etc. However, all these techniques can be considered as modifications of the general skeleton [Khaled A. Al-Utaibi, 2010], shown in Fig. 1.

In this paper the original algorithm proposed in [Khaled A. Al-Utaibi, 2010] is implemented and a modification to it is proposed. The original algorithm is based on implementation of a chaotic-based image encryption using two bio-operations, multi-point crossover and mutation, as tools for confusion and diffusion. The proposed modification is to scramble the pixel values in the mutated image by using a random sequence. The modification is proposed because the original algorithm is not immune to differential attacks. The remainder of this paper is organized as follows: In Section 2, detailed description of the proposed modified image encryption algorithm is given. Experimental results in Section 3 demonstrate various performance and security measures of the algorithm and a comparison of the values obtained after implementation of the original algorithm and the values obtained after implementation of the modified algorithm is also presented. Section 4 concludes the paper by summarizing the proposed work and the obtained results.

2. The proposed algorithm

2.1 The General Structure of the Algorithm

The general structure of the proposed encryption algorithm is shown in Fig. 2. It consists of a logistic map, a quantification unit, a crossover unit, a mutation unit and a scrambler.

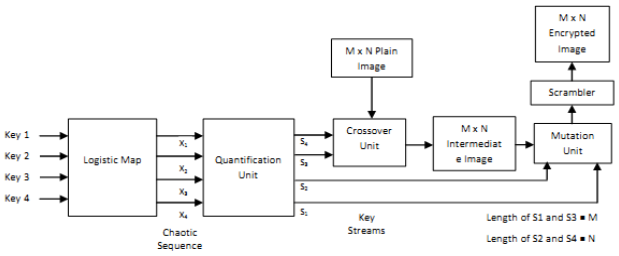

Fig.2 General structure of the proposed algorithm

Since the modification to the algorithm is made only in the last stage, the initial blocks and their working remains the same as in [Khaled A. Al-Utaibi, 2010]. Based on the given controlling parameters (μ_1, μ_2, μ_3, μ_4) and initial values (x_{10}, x_{20}, x_{30}, x_{40}) which represent the set of shared keys used by encryption/decryption algorithm, the logistic map generates four chaotic sequences. The quantification unit maps the chaotic sequence to four key streams (S_1, S_2, S_3, S_4) which are used in controlling the operation of the crossover and mutation units. The purpose of the crossover unit is to change the order of the image pixels row-wise and column-wise (image confusion). The mutation unit is used to mask the intermediate image obtained by the crossover unit with a random image (image diffusion). The Scrambler alters the position of the image pixels using a random sequence.

2.2 Logistic Map

Logistic map is widely used in chaotic cryptography for their simplicity and high sensitivity to initial conditions. It is defined as

$$a_{n+1} = r a_n (1 - a_n), \quad 0 \le a \le 1 \qquad [1]$$

where μ is a control parameter, x_n is a real number in the range [0,1] and x_0 is an initial condition. When $3.569955672 < \mu \le 4$, the system becomes chaotic [Y. Feng, J, 2009]. In the proposed algorithm, the logistic map is used to generate four chaotic sequences (X_1, X_2, X_3, X_4). The four sequences are generated based on some given controlling parameters (μ_1, μ_2, μ_3, μ_4) and initial values (x_{10}, x_{20}, x_{30}, x_{40}) which are considered as the shared keys. These chaotic sequences are used by the quantification unit to generate the four key streams required by the crossover and mutation units.

2.3 The Quantification Unit

Most of chaotic systems generate real-valued sequences which need to be mapped to integer/binary sequences (i.e. key streams) which will be used to control the confusion and diffusion units. Basically, there are three techniques commonly used in the literature: normalization, threshold level functions, and ordered chaotic sequence.

Ordered chaotic sequence method, as described in [Y. Feng. J, 2009], is based on mapping the key stream to the element's positions in the sorted chaotic sequences. In this method, the elements in the chaotic sequence, X, are sorted in ascending order to form an ordered sequence X'. If the chaotic sequences are non-periodic, then each element in X has exactly one position in the sorted sequence X'. These positions are taken to be the values of the key stream. For example, suppose that X = {0.87, 0.34, 0.12, 0.75, 0.03, 0.88, 0.56, 0.04}, then the sorted sequence X' = {0.03, 0.04, 0.12, 0.34, 0.56, 0.75, 0.87, 0.88}. Since each element in X has exactly one position in X' (e.g. 0.87 has position 7), the key stream is given by S = {7, 4, 3, 6, 1, 8, 5, 2}. This method is used in the proposed algorithm for its simplicity and short computation time.

The quantification unit in the proposed algorithm receives four chaotic sequences (X_1, X_2, X_3, X_4) generated by the logistic map and converts them to four key streams (S_1, S_2, S_3, S_4) which will be used to control the operation of the crossover and mutation units. The length of the 1st and 3rd key streams is M, and the length of the 2nd and 4th key streams is N, where M × N is the size of the plain image in pixels.

2.4 The Crossover Unit

The crossover unit is used to change the order of the image pixels row-wise and column-wise by means of a multi-point crossover operation. The unit is controlled by the two key streams, S1 and S2, which are generated by the chaotic map and the quantification unit. The first key stream controls the crossover operation on the image rows, while the other controls the crossover operation on the image columns. Each pair of the two consecutive elements in the key stream selects two rows/columns for the crossover operation and determines the positions of the cut points. The number of cut points in the crossover operation is a variable parameter that can be set by the user prior to encryption/decryption process. For example, this value can be set to M/2 for row-crossover and N/2 for column-crossover. The idea of selecting the two rows/columns and determining the positions of the cut points can be explained as follows. Assume that the two consecutive elements of the key stream are E_i and E_{i+1}, then rows/columns number E_i and E_{i+1} are selected for crossover operation. The positions of the cut points are computed as follows:

$$r_1 = \max\{1, |Ei - Ei + 1| modL\}$$
$$r_2 = \max\{1, r_1 + |Ei - Ei + 1| modL\}$$
$$\ldots\ldots\ldots$$
$$r_k = \max\{1, r_{k-1} + |Ei - Ei + 1| modL\} \quad [2]$$

where k is the number of cut points, (r_1, r_2, …, r_K) are the positions of the K cut points, and L is the length of the row/column (i.e. L = M or N). For example, assume that the number of cut points is 4 and the values 8 and 5 are two consecutive elements in the key stream S1. Then, the 5th and 8th rows will be selected, and the positions of the cut points will be determined as shown in Fig. 3. The computation of the positions of the cut points can be optimized, if we compute the set (r_1, r_2, …, r_K) in advance based on all possible values of $|E_i - E_{i+1}|$ and store them in a table referenced by $|E_i - E_{i+1}|$. After selecting two rows/columns, i and j, and determining the positions of the cut points r_1, r_2, …, and r_K, the multi-point crossover operation is performed by swapping the even segments of i and j as shown in Fig. 4. The set of the cut points' positions are sorted in an ascending order. Also it is possible to swap odd segments instead of even ones.

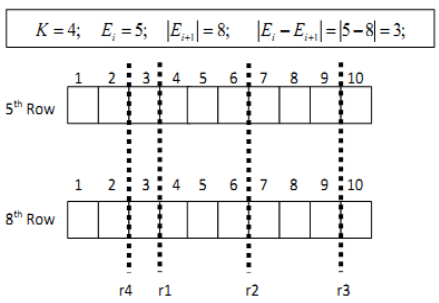

Fig.3 Example to determine the positions of cut points (before sorting the values of r_1, r_2, r_3, and r_4)

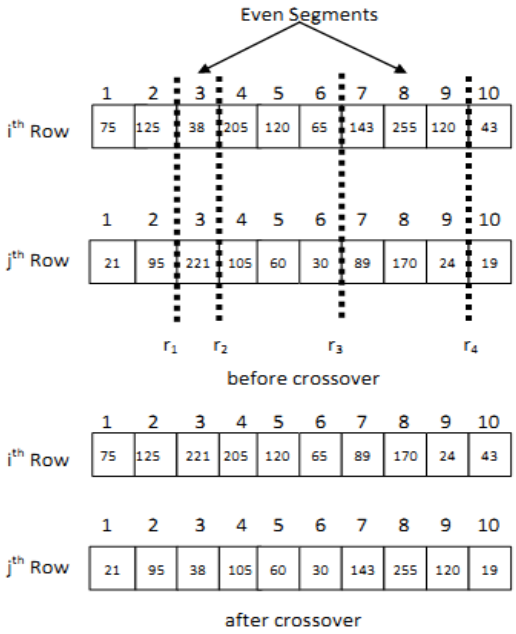

Fig.4 Example to perform the crossover operation

2.5 The Mutation Unit

The purpose of the mutation unit is to mask the intermediate image obtained by the crossover unit with a random image using XOR operation to obtain the encrypted image. For this purpose the sender and receiver must first agree on some randomly generated image and keep it secret. Then, the mutation unit XORs every pixel in the intermediate image with pseudo-random pixel from the secrete image selected by the values of the two key streams S_3 and S_4. For instance, the (i, j) th pixel in the encrypted image is obtained by XORing the corresponding pixel in the intermediate image with (p_i, q_j)th pixel of the secret image, where p_i ϵ S_3 and q_i ϵ S_4.

2.6 Scrambler

The purpose of the scrambler is to scramble the positions of the pixel values of the mutated image. To serve the purpose the sender and receiver must agree to use the same random number generator (RNG). The sender and the receiver generate the same scrambling sequence using the same seed and the same RNG. The scrambling operation is carried out changing the positions of the pixel values of the mutated image as per elements in random sequence. It is illustrated as follows in table 1.

Table 1 Scrambler unit design

	Random Sequence					
Random Sequence		2	3	1	…	N
	2	(2, 2)	(2, 3)	(2, 1)		(2, N)
	3	.	.	.		.
	1	.	.	.		.
	.	.	.	.		.
	N	(N, 2)	(N, 3)	(N, 1)	…	(N, N)

with (p_i, q_j)th pixel of the secret image, where $p_i \in S_3$ and $q_i \in S_4$.

2.7 Operation of the Proposed Encryption Algorithm

Given an M×N image, the general operation of the proposed encryption algorithm is described as follows:

Step 1. Using the logistic map with key values (μ_1, x_{10}, μ_2, x_{20}, μ_3, x_{30}, μ_4, x_{40}) generate four chaotic sequences X_1, X_2, X_3 and X_4 where lengths of X_1 and X_3 = M and lengths of X_2 and X_4 = N.

Step 2. Using sorted chaotic sequence method, obtain four key streams S_1, S_2, S_3 and S_4 where lengths of S_1 and S_3 = M and lengths of S_2 and S_4 = N.

Step 3. Perform crossover operation row-wise using the key stream S_1.

Step 4. Perform crossover operation column-wise using the key stream S_2.

Step 5. Perform mutation operation by XORing each pixel in the intermediate image obtained by the crossover operation with a random pixel selected from the secret random image based on the key streams S_3 and S_4.

Step 6. Perform scrambling operation by changing the positions of the pixel values of the mutated image as per the secret random sequence.

Step 7. For color image encryption repeat steps 3, 4, 5 and 6 for all the three planes viz., Red, Green and Blue.

3. Experimental results

The algorithm can be applied to grayscale as well as color images. To test the performance of the proposed modified technique, a number of experiments using MATLAB 7.7.0 (R2008b) are carried out. These experiments include image encryption and decryption, histogram analysis of the plain and encrypted images, key space and sensitivity analysis, correlation coefficient analysis, Entropy and NPCR and UACI test. To generate mutation image and the scrambling sequence 'rand' and 'randiperm' commands in MATLAB are used respectively. In addition, the proposed algorithm is compared with the original algorithm in terms of average correlation coefficients of the encrypted image, algorithm run time in seconds, NPCR, UACI and Entropy.

3.1 Image Encryption and Histogram Analysis

For experiments on grayscale image, a 256 × 256 size gray scale cameraman image shown in Fig.5 (a) is used. This image was encrypted using the proposed technique with a key = {3.7158, 0.11, 3.89858, 0.25, 3.76158, 0.35, 3.8458, 0.6520}. The resulting encrypted image is shown in Fig. 5(b).

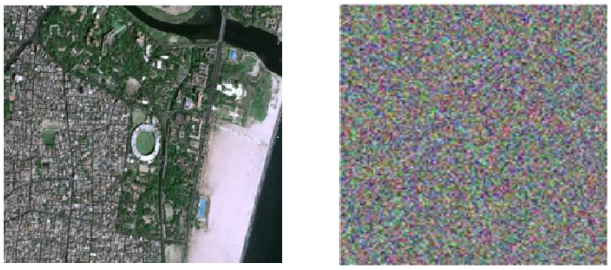

Fig.5a The original plain image (256 × 256 size cameraman image) **b**. The Encrypted Image

The histograms of plain image and encrypted image are shown in Fig. 6. It is clear from these figures that the histogram of the encrypted image is uniform and is different from the histogram of the plain image. This makes it hard for cryptanalysis.

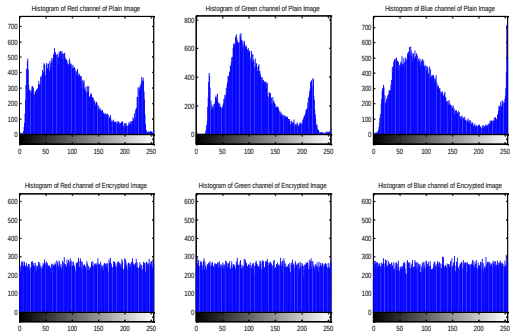

Fig.6 Histogram of Plain Image and Encrypted Image

3.2 Key Space and Sensitivity Analysis

Since the modified algorithm makes use of the same key as the original algorithm, the secret key and the related analysis hold good for the modified algorithm also. The secret key is (μ_1, x_{10}, μ_2, x_{20}, μ_3, x_{30}, μ_4, x_{40}), where $\mu_i \in 3.569945672 \ldots 4$, and $x_i \in [0,1], i = 1,2,3,4$, μ_i and x_{i0} are both double precision. Since double precision can represent about 16 decimal digits, the key space of the proposed algorithm can be estimated as $(10^{14})^4 \times (10^{16})^4 = 10^{120} \approx 2^{398}$. Note that the range of μ_i is [3.569945672…, 4]; therefore a 14-digit precision is assumed. Thus, brute-force attacks on the key are computationally infeasible.

The brute-force attacks on the key streams, S_1, S_2, S_3 and S_4, generated by the quantification unit is also computationally infeasible as there are $L_i!$ combinations for each sequence, where L_i is the length of each sequence (i = 1, 2, 3, 4). Note that when these sequences are considered together to control the crossover and mutation operations, then the total possible combinations become $(L_1 \times L_2 \times L_3 \times L_4)! = M^2 \times N^2$.

A key sensitivity test was carried out using a key that is one digit different from the original key to decrypt the encrypted image. The resulting image is totally different from the original image as shown in Fig. 8. This demonstrates that the proposed algorithm is very sensitive to any change in the secret key value.

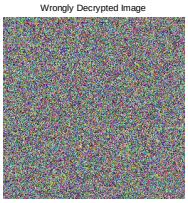

Fig.8 Image decrypted with a wrong key of one digit difference from the original key.

3.3 Correlation of Two Adjacent Pixels

In this experiment, the correlation between two adjacent pixels in the plain image and encrypted image is tested. The following equations have been used to calculate the correlation coefficients r(x,y) in horizontal (HC), vertical (VC) and diagonal (DC) directions:

$$r(x,y) = \frac{COV(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad [3]$$

where x and y are gray scale values of two adjacent pixels in the image, D(x) is the variance of x and COV(x, y) is the covariance of x and y. The experiment was performed by randomly selecting 4096 pairs of adjacent pixels from the plain image and the encrypted image, and then calculating the correlation coefficients using (3) – (6). Horizontal, Vertical and Diagonal Pixel Distribution of Plain Color Image for red, green and blue channels are shown in Fig. 9(a) and that for encrypted image is shown in Fig. 9(b). The correlation coefficients are documented in Table 2.

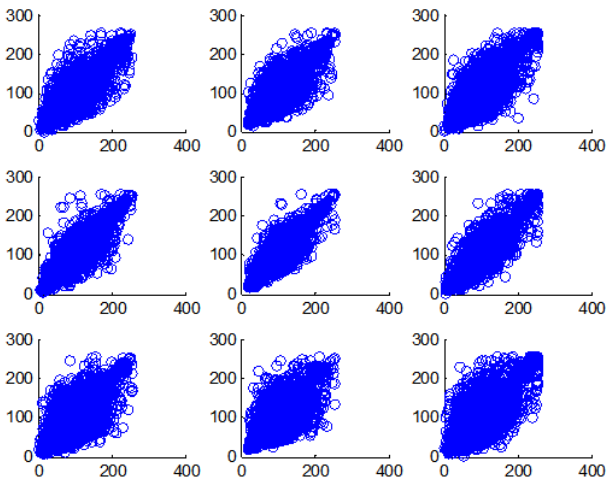

Fig.9a Horizontal, Vertical and Diagonal Pixel Distribution of Plain Image

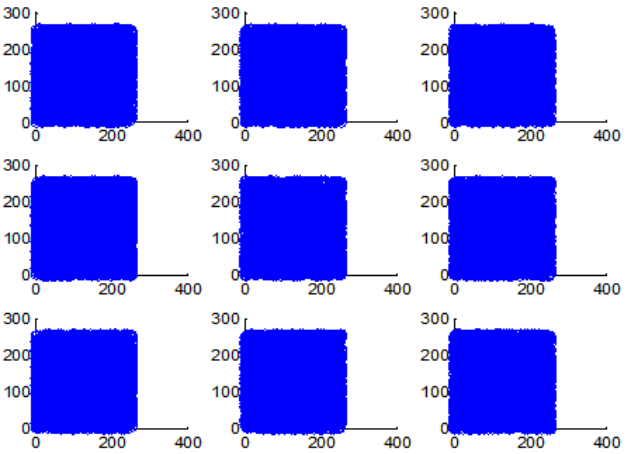

Fig.9b Horizontal, Vertical and Diagonal Pixel Distribution of Cipher Image

Table 2 Correlation coefficients of two adjacent pixels in plain and encrypted image

	Plain Image			Encrypted Image		
Color	*R*	*G*	*B*	*R*	*G*	*B*
HC	0.9545	0.9491	0.9223	-0.0005	-0.0008	0.0027
VC	0.9680	0.9558	0.9542	-0.0006	-0.0266	0.0052
DC	0.9269	0.9034	0.8875	-0.0242	-0.0109	-0.0051

3.4 Differential Analysis

The number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI) are two most common quantities used to evaluate the strength of image encryption algorithms/ciphers with respect to differential attacks. NPCR means the change rate of the number of pixels of the cipher-image when only one pixel of the plain-image is modified. The UACI index measures the average intensity of differences between two images. Let us assume two cipher pictures C_1 and C_2 with one-pixel difference in their corresponding plain images. The pixel values of ciphered images C_1 and C_2 at row i and column j are labeled as $C_1(i, j)$ and $C_2(i, j)$ respectively. NPCR and UACI are defined as:

$$NPCR = \sum_{i,j} \frac{D(i,j)}{M \times N} \times 100\% \quad [4]$$

$$UACI = \sum_{i,j} \frac{|C_1(i-j) - C_2(i,j)|}{M \times N \times 256} \quad [5]$$

Where

$$D(i,j) = 0, if\ C_1(i,j) = C_2(i,j) \quad [6]$$

And

$$D(i,j) = 1, if\ C_1(i,j) \neq C_2(i,j) \quad [7]$$

The results for a 256 × 256 satellite image shown in Figure 5(a) are given in table 3 that shows the method is resistive to differential attacks.

Table 3 NPCR, UACI and entropy results

	Encrypted Image		
Color	*R*	*G*	*B*
NPC (%)	99.6368	99.6017	99.6201
UACI (%)	33.5262	33.6828	33.5840
Entropy (bits)	7.9974	7.9974	7.9973

3.5 Information Entropy Analysis

It is well known that the entropy H(s) of a message source m can be calculated as

$$H(m) = \sum_{i=1}^{2N-1} P(m_i) \log_2 \left(\frac{1}{P(m_i)}\right) \quad (bits) \qquad [8]$$

where $P(m_i)$ represents the probability of symbol mi and log represents the base 2 algorithm so that the entropy is expressed in bits. Assume that there are 28 states of the information source and they appear with the same probability, according to Eq.(8), we can get the ideal H(m) = 8, which shows that the information is random. Hence the information entropy of the ciphered image should be close to 8 after encryption. If entropy is less than 8, there is possibility certain degree of predictability. Result of entropy for encrypted image shown in Fig. 5 (b) is H(m) = 7.9973 bits ≈ 8 bits which corresponds to a true random source.

3.6 Comparison with original algorithm

The values of the various parameters like Correlation coefficients, NPCR, UACI, Entropy and Encryption time, obtained after performing different experiments are compared with the original algorithm and tabulated in Table 4. Since in the original algorithm grayscale Lena image is considered, respective results with the same image are presented and compared.

Table 4 Comparison of results with original algorithm

Parameters	Original Algorithm		Modified Algorithm	
Grayscale Image	*Plain Image*	*Encrypted Image*	*Plain Image*	*Encrypted Image*
HC	0.9392	-0.0026	0.9400	-0.0016
VC	0.96526	-0.0059	0.9595	0.0010
DC	0.9104	0.0196	0.9279	-0.0017
NPCR (%)	0.0015%		99.6414%	
UACI (%)	0.0006%		33.4265%	
Entropy (bits)	7.9970 bits		7.9970	
Time (sec)	0.305977 seconds		0.5132 seconds	

Encryption time required for to generate the cipher images of the grayscale and color plain images by applying the algorithm to them is tabulated in Table 5.

Table 5 Encryption time

Image and Image size in pixels	Time in seconds
256 × 256 Grayscale Image	0.5132 seconds
256 × 256 Color Image	0.8211 seconds

Conclusion

In this paper, a modified bio-inspired image encryption algorithm based on chaotic maps is proposed. The algorithm uses a logistic map to generate four chaotic sequences which are converted to four key streams using sorted chaotic sequences method. The generated key streams are used to control a multi-point crossover operation which represents the confusion part of the encryption algorithm and a mutation operation which represents the diffusion part. The proposed modification i.e., the scrambling operation adds to diffusion and makes the algorithm immune to differential attacks. Experimental results show that the proposed algorithm is capable of generating encrypted images with uniform distribution of the pixel values, very low correlation coefficients of adjacent pixels, completely random with Entropy = 7.9973 bits and is very sensitive to secret key values. The Cipher image generated by this algorithm is free from threat posed by differential attacks with NPCR = 0.996 and UACI = 0.335. The encryption time is as low as about 500 milliseconds for grayscale image and 2 seconds for color image. Due to addition of an extra step in the algorithm the encryption time is 200 milliseconds more compared to the original algorithm, but it ensures immunity towards differential attacks as visible from the NPCR and UACI values.

References

Sahar Mazloom, Amir-Masud Eftekhari-Moghadam (April 2011), "Color Image Cryptosystem using Chaotic Maps", in Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP), IEEE Symposium, pp. 142-147.

Rashidah Kadir, Rosdiana Shahril, Mohd Aizaini Maarof (May 2010), "A modified image encryption scheme based on 2D chaotic map", in Computer and Communication Engineering (ICCCE), International Conference, pp. 1-5.

Z. Lin and H. Wang (July 2009), "Image encryption based on chaos with PWL memristor in Chua's circuit", in Proc. of the International Conference on Communications, Circuits and Systems, (ICCCAS, pp. 964-968.

K. C. Ravishankar, M. G. Venkateshmurthy (June 2006), "Region Based Selective Image Encryption", in International Conference Computing & Informatics(ICOCI '06, pp. 1-6.

Aihong Z, Lian L, Shuai Z (Oct 2010), "Research on Method of Color Image Protective Transmission Based on Logistic Map", in Computer Application and System Modeling (ICCASM), 2010 International Conference, Vol. 9, pp. V9-266 - V9-269.

Chen Gang, Zhao Xiao-Yu, Li Jun-Li (2005), "A Self-Adaptive Algorithm on Image Encryption", in Journal of Software, Vol.16(11): pp. 1975–1982.

Xiaofeng Liao, Shiyue Lai, Qing Zhou, "A novel image encryption algorithm based on self-adaptive wave transmission", in Signal Processing, 2010, pp. 2714 –2722.

Shujiang Xu, Yinglong Wang, Jizhi Wang, Yucui Guo (July 2010), "A Fast Image Encryption Scheme Based on a

Nonlinear Chaotic Map", in Signal Processing Systems (ICSPS), 2nd International Conference, Volume 2, pp. V2-326 - V2-330.

Liu Hongjun, Wang Xingyuan (2010), "Color image encryption based on one-time keys and robust chaotic maps", in Computers and Mathematics with Applications, pp. 3320-3327.

Khaled A. Al-Utaibi, El-Sayed M. El-Alfy (July 2010), "A Bio-Inspired Image Encryption Algorithm Based on Chaotic Maps", in Evolutionary Computation (CEC), IEEE Congress, pp. 1-6.

Y. Feng, J. Li and X. Yang (Aug 2009), "Discrete chaotic based 3D image encryption scheme," in Proc. of the Symposium on Photonics and Optoelectronics, (SOPO), pp. 1-4.