An Encryption Scheme for Color Images Based on Chaotic Maps and Genetic Operators

El-Sayed M. El-Alfy
College of Computer Sciences and Engineering
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia
alfy@kfupm.edu.sa

Khaled A. Al-Utaibi
Information and Computer Science Department
University of Ha'il
Ha'il, Saudi Arabia
alutaibi@uoh.edu.sa

Abstract—**Secure transmission and storage of color images is gaining growing importance in recent years due to the proliferation of multimedia network applications. In this paper, we propose a novel scheme based on chaotic maps and genetic operators for encrypting color images. The capability of the proposed approach to efficiently generate cipher images with very low correlation coefficients of adjacent pixels is demonstrated through some experimental results for several benchmark images. It is also shown that the approach is very sensitive to any slight changes in the secret key values.**

Keywords-image encryption; chaotic maps; genetic operator; information hiding; data security.

I. INTRODUCTION

Color images are being transmitted and stored heavily over the Internet and wireless networks taking advantage of rapid development in multimedia and network technologies. However, as there is always a potential risk of information security in such interconnected environments, protecting confidentiality of color images has become an increasingly important issue in many areas such as remote sensing and satellite imagery, astrophysics, seismology, agriculture, radiology, telemedicine, ecosystems, industrial processes, military communications, and image archiving. Several image encryption schemes have been suggested in the literature to meet this requirement [1][2]. However, due to the processing overhead resulting from the large data size of digital images and the high correlation among pixels, traditional encryption techniques, such as DES, AES and RAS, are found to be inefficient for image encryption [7][8][11].

The pseudorandom nature and other properties of chaotic systems, including sensitivity to initial conditions and non-periodicity, have made them attractive alternatives among the proposed approaches for image encryption [7]. The first chaotic based image encryption algorithm was proposed in 1989 [17]. Recently, there is a growing interest in this area and several approaches have been proposed in the literature [3]-[6]. In [7], Lin and Wang proposed an encryption algorithm based on chaos with PWL memristor in Chua's circuit. Their algorithm uses two main operations of image scrambling and pixel replacement. Fu and Zhu proposed another technique based on logistic maps with permutation and circular bit-shift methods for confusion and diffusion [8]. The method proposed by Yanling is based on logistic chaotic sequences and image mirror mapping [9]. A 3D image encryption scheme using logistic maps with bit permutation was presented in [10]. The scheme proposed by Kumar and Chandrasekaran is also a 3D image encryption, but with a different approach where Lorenz attractor is used directly for image encryption [11]. Lue *et al.* proposed an image encryption algorithm based on spatiotemporal chaos [12]. In their work, the plain image block data is masked by the values extracted from a spatiotemporal chaotic system, and then shuffled according to the maximum state value in the system. Wei-Bin and Xin proposed an algorithm that uses Arnold cat map to shuffle the pixels of the plain image and 1D Henon's chaotic system to change the shuffled pixels by XOR operation [13]. The algorithm proposed by Wang and Zhang is based on S-boxes in AES and chaotic sequences generated by logistic maps [14]. The algorithm presented by Flores-Carmona *et al.* in [15] is based on CML (Chaotic Map Lattice), which allows direct encryption and decryption of color digital images. A 3D Bakeer map encryption technique was proposed by Hongelei and Guang-Shou in [16]. Chong Fu *et al.* proposed an image encryption scheme based on 3D Lorenz system to improve the security and performance of the encryption system over conventional one dimension chaos based ones [18].

The previously mentioned algorithms are restricted to grayscale images. Though, some of them can be easily extended to handle color images, this extension comes with a cost of increased computation time as a result of additional information required to represent color components. Therefore, many color-image encryption techniques use block-based encryption which is usually faster than stream-based encryption although it may be less secure. One example of block-based encryption algorithm for colored images was proposed by Pareek *et al.* [19]. This algorithm uses an external key and two logistic maps. The first map is used to generate the initial conditions of the second map which is used to select the type of encryption operation among eight different encryption operations (*e.g.*, NOT, XOR, etc.). In order to make the cipher robust against attacks, the external key is modified after encrypting each block of pixels. The color image encryption algorithm proposed by Shubo *et al.* in [20] uses two logistic maps coupled such that the first logistic map updates the parameter of the other. The encryption operation is performed by a simple XOR operation of the binary sequence of the plain-image with the keystream binary sequence generated by the second logistic map. Another color image encryption based on a modified logistic map and 4-dimensional hyper-chaotic maps was proposed in [21]. In this, paper, we propose an alternative scheme for encrypting color images based on

ISBN: 978-1-61208-133-5

chaotic-maps and genetic operations as tools for confusion and diffusion. The simple and fast computation of crossover and mutation operations compared to regular confusion and diffusion operations allows the algorithm to implement stream-based encryption which usually provides better security than block-based encryption.

The rest of this paper is organized as follows. In Section II, we give a detailed description of the proposed image encryption algorithm. Experimental results in Section III demonstrate various performance and security measures of our algorithm. Section IV concludes the paper by summarizing the proposed work and the obtained results.

II. THE PROPOSED ALGORITHM

A. *The General Structure of the Algorithm*

The general structure of the proposed algorithm is shown in Figure 1. It consists of four units: logistic map, quantification, crossover, and mutation. The logistic map generates four chaotic sequences based on the given controlling parameters (μ_1, μ_2, μ_3, μ_4) and initial values (x^o_1, x^o_2, x^o_3, x^o_4) which represent shared keys used by the encryption and decryption algorithms. The quantification unit maps the four chaotic sequences to four key streams which are then used to control the crossover and mutation operations. The purpose of the crossover unit is to cause image confusion by scrambling the image pixels row-wise and then column-wise. The mutation unit is used to mask the intermediate image obtained by the crossover unit with a random image; thus causing image diffusion.

B. *Logistic Map*

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

$$x_{n+1} = \mu x_n (1 - x_n), \tag{1}$$

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 < $\mu \leq 4$, the system becomes chaotic [10].

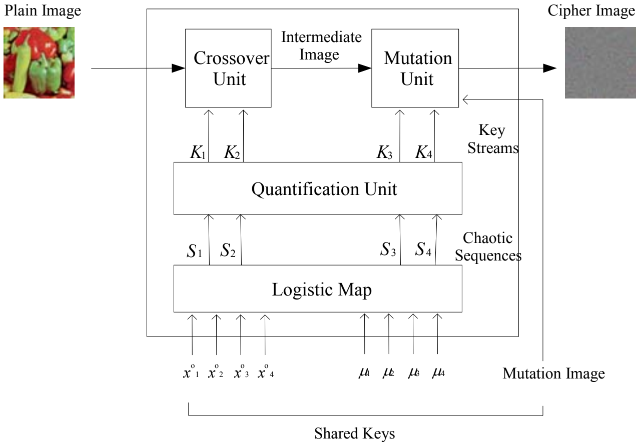


Figure 1. Layout of the proposed algorithm.

In the proposed algorithm, the logistic map is used in a similar manner as in [8] to generate four chaotic sequences (S_1, S_2, S_3, S_4). These sequences are generated based on some given controlling parameters (μ_1, μ_2, μ_3, μ_4) and initial values (x^o_1, x^o_2, x^o_3, x^o_4) which are considered as shared keys for encryption and decryption.

C. *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 literatures: normalization, threshold level functions, and ordered chaotic sequence.

In normalization methods, a real value, X, in the chaotic sequence can be mapped to a digital value, D, in the key stream using the following relation:

$$D = \left\lceil \frac{(X - X_{\min}) \times D_{\max}}{X_{\max} - X_{\min}} \right\rceil, \tag{2}$$

where $X_{\min}$ and $X_{\max}$ are the minimum and maximum values in the chaotic sequence to be quantified, and $D_{\max}$ is the maximum required value of the key stream.

In the second method, each value, x_i, in the chaotic sequence is converted to a binary bit, b_i, using a single level threshold function defined as:

$$b_i = \begin{cases} 0 & x_i < 0.5 \\ 1 & x_i \geq 0.5 \end{cases}. \tag{3}$$

The third method as described in [10] 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 the sequence K = {7, 4, 3, 6, 1, 8, 5, 2}.

The first method is a simple and fast way to map the chaotic sequences to integer key streams, but it is subject to rounding errors. The threshold level function gives uniform distribution of the generated key streams. However, it is a lengthy process and requires long chaotic sequences (each bit requires one chaotic value). The third method is used in the proposed algorithm for its simplicity and short computation time.

The quantification unit in our algorithm receives four chaotic sequences (S_1, S_2, S_3, S_4) generated by the logistic map and convert them to four key streams (K_1, K_2, K_3, K_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 $M \times N$ is the size of the plain image in pixels.

D. 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, K_1 and K_2, generated by the chaotic map and quantification units. The first key stream controls the crossover operation on the image rows whereas the other key controls the crossover operation on the image columns. Each two consecutive elements in the key stream select two rows/columns for the crossover operation and determine the positions of the cut points. The number of cut points in the crossover operation is a variable parameter that should be set by the user prior to encryption/decryption process. For example, this value can be set to $\lfloor M/2 \rfloor$ for row-crossover and $\lfloor N/2 \rfloor$ 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:

$$\begin{array}{lcl} r_1' & = & |E_i - E_{i+1}| \bmod L \\ r_2' & = & (r_1' + |E_i - E_{i+1}|) \bmod L \\ \vdots & & \\ r_P' & = & (r_{P-1}' + |E_i - E_{i+1}|) \bmod L \\ (r_1, & r_2, & \ldots, \ r_P) = sort(r_1', \ r_2', \ \ldots, \ r_P') \end{array} \tag{4}$$

where P is the number of cut points, (r_1, r_2, …, r_P) are their positions, and L is the length of the row (or column), *i.e.*, $L = M$ (or N). Note that the sort procedure rearranges the values of the temporary variables in ascending order. For example, assume that the number of cut points is 4 and two consecutive elements in the key stream K_1 are 5 and 8. Then, the 5th and 8th rows will be selected, and the positions of the cut points will be determined as shown in Figure 2.

The computation of the positions of the cut points can be optimized, if the set (r_1, r_2, …, r_P) is computed in advance based on all possible values of $|E_i - E_{i+1}|$ and store them in a lookup table referenced by $|E_i - E_{i+1}|$. After selecting two rows (or columns), i and j, and determining the positions of the cut points r_1, r_2, …, and r_P, the multi-point crossover operation is performed by swapping RGB of pixels in the even segments of the two rows (or columns) i and j as shown in Figure 3. Note that, it is possible to swap odd segments instead of even ones.

E. The Mutation Unit

The mutation unit is the last stage in the encryption process. To obtain the final cipher image, the mutation unit masks the intermediate image resulting from the crossover stage with a random image using XOR operation. 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 K_3 and K_4. For instance, the (i, j)th pixel in the cipher image is obtained by XORing the corresponding pixel in the intermediate image with (p_i, q_j)th pixel of the secret image, where $p_i \in K_3$ and $q_i \in K_4$. This process is explained further by means of a simple example of 4×4 image as shown in Figure 4.

F. Operation of the Proposed Encryption Algorithm

Given an RGB color image, where each one of the three color components (*i.e.*, red, green and blue) is represented as an $M \times N$ matrix, the general operation of the proposed encryption algorithm is described as follows:

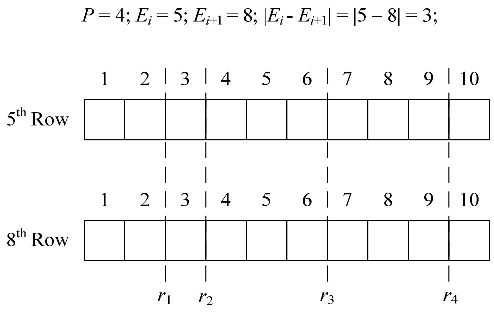
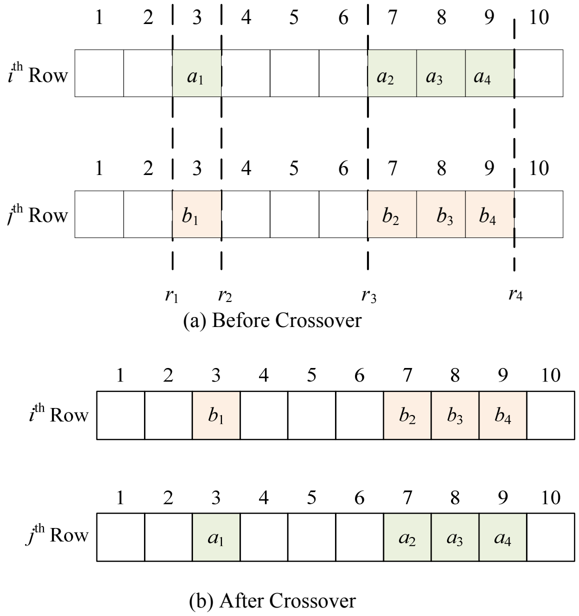


Figure 2. Example of determining the positions of cut points.



(a) Before Crossover

(b) After Crossover

Figure 3. Example of the crossover operation.

Step 1: Using the logistic map with key values (μ_1, x^{o}_1, μ_2, x^{o}_2, μ_3, x^{o}_3, μ_4, x^{o}_4) generate four chaotic sequences S_1, S_2, S_3 and S_4 where $|S_1| = |S_3| = M$ and $|S_2| = |S_4| = N$.

Step 2: Using sorted chaotic sequence method, obtain four key streams K_1, K_2, K_3 and K_4 where $|K_1| = |K_3| = M$ and $|K_2| = |K_4| = N$.

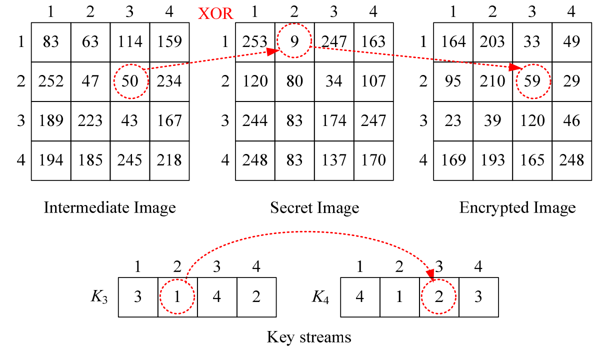
Step 3: Perform crossover operation row-wise on each individual $M \times N$ matrix using the key stream K_1.

Step 4: Perform crossover operation column-wise on each individual $M \times N$ matrix using the key stream K_2.

Step 5: Perform mutation operation on each individual

ISBN: 978-1-61208-133-5

matrix by XORing each pixel in the intermediate matrices obtained by the crossover operation with a random pixel selected from corresponding matrix in the secret image based on the key streams K_3 and K_4. The secret masking image as mentioned previously is generated randomly and shared by the sender and the receiver.



Intermediate Image

	1	2	3	4
1	83	63	114	159
2	252	47	50	234
3	189	223	43	167
4	194	185	245	218

Secret Image

	1	2	3	4
1	253	9	247	163
2	120	80	34	107
3	244	83	174	247
4	248	83	137	170

Encrypted Image

	1	2	3	4
1	164	203	33	49
2	95	210	59	29
3	23	39	120	46
4	169	193	165	248

Key streams

	1	2	3	4
K_3	3	1	4	2
K_4	4	1	2	3

Figure 4. Example of the mutation operation of one color component of 4×4 block.

G. *Decryption Algorithm*

The decryption algorithm is identical to the encryption algorithm discussed above except that the order of the basic operations is reversed. That is, after generating the required key streams in steps 1 and 2, the decryption algorithm applies mutation operation first followed by column wise crossover operation then row-wise crossover operation.

III. Experimental Results

To empirically assess the effectiveness of the proposed technique, we have carried out a number of experiments using MATLAB 7.7.0 (R2008b). These experiments include image encryption and decryption, histogram analysis of the plain and encrypted images, key space and sensitivity analysis and correlation coefficient analysis.

A. *Image Encryption and Histogram Analysis*

For this experiment, we have considered a 24-bit color image of size 256×256 pixels shown in Figure 5 (a), which is available at USC-SIPI image database in TIFF format [22].

This image is encrypted using the proposed technique with a key = {3.7158, 0.11, 3.89858, 0.25, 3.76158, 0.35, 3.8458, 0.552}. The resulting encrypted image is shown in Figure 5 (b). The histograms of red, green and blue channels of the plain and the encrypted images are shown in Figure 6. It is clear from this figure that the histograms of the encrypted image are uniform and significantly different from the histograms of the plain image. This result indicates that it is very difficult to use statistical analysis to attack the proposed encryption algorithm.

B. *Key Space and Sensitivity Analysis*

The secret key of the proposed technique is (μ_1, x^o_1, μ_2, x^o_2, μ_3, x^o_3, μ_4, x^o_4), where $\mu_i \in (3.569945672\ldots, 4]$ and $x^o_i \in (0,1)$, i = 1,2 3, 4, μ_i and x^o_i 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.

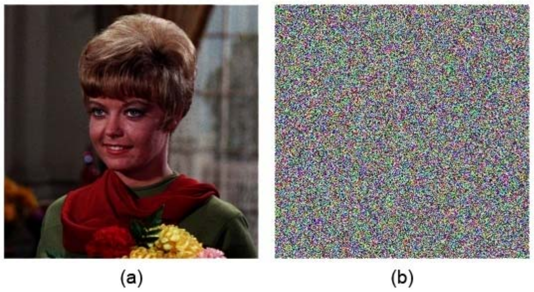


Figure 5. (a) original plain image. (b) encrypted image.

The brute-force attacks on the key streams, K_1, K_2, K_3, and K_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 $(L_1 \times L_2 \times L_3 \times L_4)! = (M^2 \times N^2)!$.

We have carried out a key sensitivity test 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 Figure 7. This demonstrates that the proposed algorithm is very sensitive to any change in the secret key value.

C. *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 formula [19] has been used to calculate the correlation coefficients in horizontal and vertical directions:

$$C_r = \frac{N\sum_{j=1}^{N}(x_j \times y_j) - \sum_{j=1}^{N} x_j \times \sum_{J=1}^{N} y_j}{\sqrt{\left(N\sum_{j=1}^{N} x_j^{\,2}\right) - \left(\sum_{j=1}^{N} x_j\right)^2 \times \left(N\sum_{j=1}^{N} y_j^{\,2}\right) - \left(\sum_{j=1}^{N} y_j\right)^2}} \qquad (5)$$

where x and y are gray scale values of two adjacent pixels in the image, and N is the total number of pixels selected from the image for calculation. The experiment was performed by randomly selecting 4096 pairs of adjacent pixels from the plain image and the encrypted image shown in Figure 5, and then calculating the correlation coefficients using (5).

The results are shown in Figure 8. Frames (a) and (b) respectively show the distribution of two horizontally adjacent pixels in the original and encrypted images. Similarly, Frames (c) and (d) show respectively the distribution of two vertically adjacent pixels in the original and encrypted images. The correlation coefficients for the two adjacent pixels in the original and encrypted images are shown in Table I. These results show clearly that the distribution of two adjacent pixels in our results is more uniform than that reported in [20].

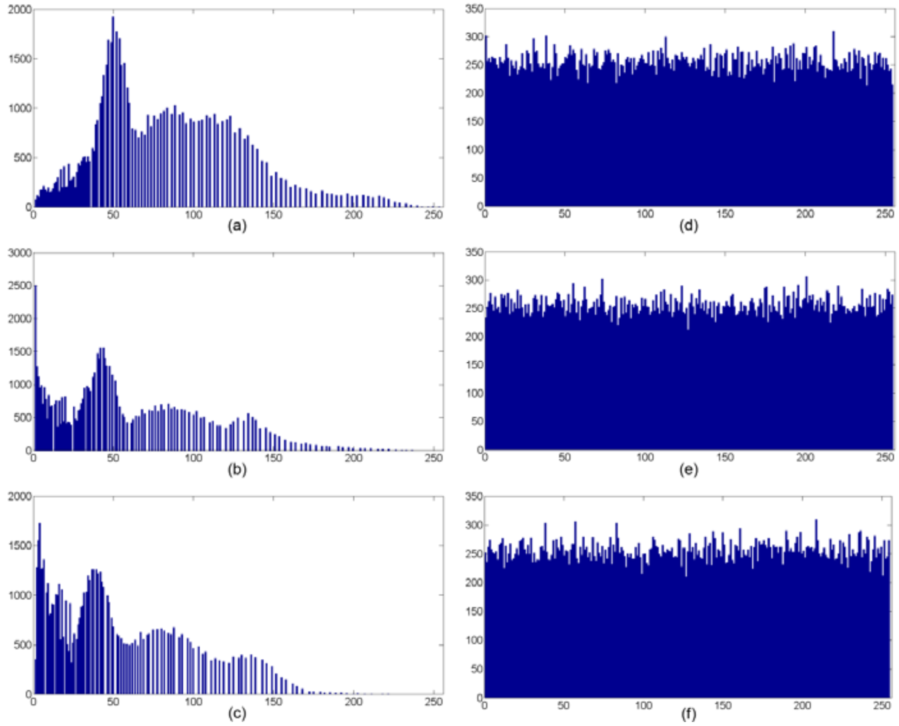


Figure 6. Histogram analysis: (a), (b), and (c) histograms of red, green and blue channels of the plain image shown in Figure 4 (a). (d), (e) and (f) histograms of red, green and blue channels of the encrypted image shown in Figure 3 (b).

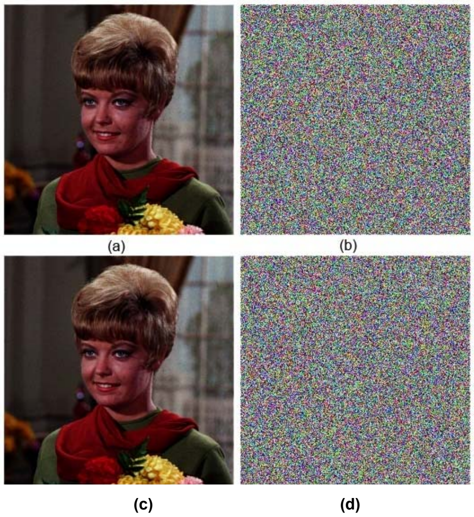


Figure 7. Key sensitivity: (a) plain image. (b) encrypted image. (c) decrypted image with key = {3.7158, 0.11, 3.89858, 0.25, 3.76158, 0.35, 3.8458, 0.552}. (d) decrypted image with key = {3.7159, 0.12, 3.89859, 0.26, 3.76159, 0.36, 3.8459, 0.553}.

TABLE I. CORRELATION COEFFICIENTS FOR TWO ADJACENT PIXELS IN PLAIN AND CIPHER IMAGES SHOWN IN FIGURE 4.

	Plain Image	Cipher Image
Horizontal	0.96784	0.00131
Vertical	0.95966	0.00012

In addition, we have carried out an extensive study of the correlation between plain image and its corresponding cipher image for several other images in the USC-SIPI image database. Results of this experiment are shown in Table II. It is clear that the correlation coefficients obtained by our proposed algorithm are very small which indicates that there is no correlation between the plain image and its corresponding encrypted image. Also, the correlation coefficients obtained by our algorithm are generally smaller than those obtained by the algorithm proposed in [20].

TABLE II. CORRELATION COEFFICIENTS BETWEEN SEVERAL PLAIN & CORRESPONDING ENCRYPTED IMAGES.

File Name	File Description	Size	Correlation Coefficient
4.1.01	Girl	256×256	-0.002601
4.1.02	Couple	256×256	-0.001354
4.1.03	Girl	256×256	0.005903
4.1.04	Girl	256×256	-0.005237
4.1.05	House	256×256	0.001596
4.1.06	Tree	256×256	-0.001793
4.1.07	Jelly beans	256×256	-0.001413
4.1.08	Jelly beans	256×256	0.002144
4.2.01	Splash	512×512	-0.000950
4.2.02	Girl (Tiffany)	512×512	-0.001311
4.2.03	Baboon	512×512	0.001832
4.2.04	Girl (Lenna)	512×512	0.000118
4.2.05	Airplane (F-16)	512×512	0.000396
4.2.06	Sailboat on lake	512×512	0.001111
4.2.07	Peppers	512×512	-0.001362
house	House	512×512	-0.000095

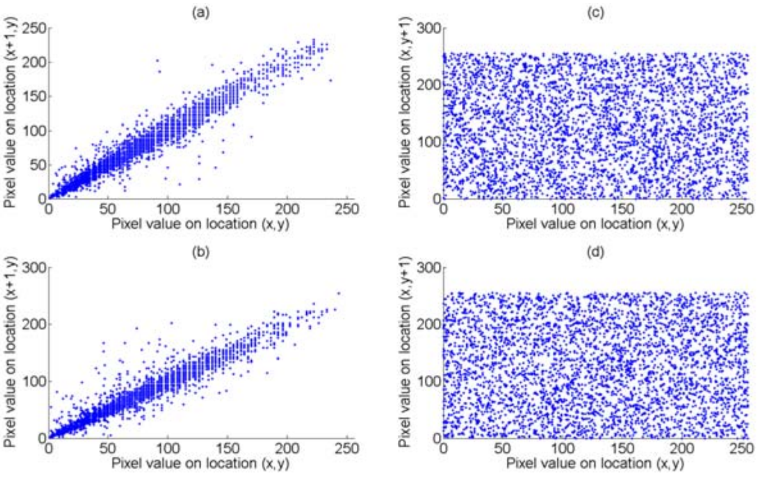


Figure 8. Correlation of two adjacent pixels: (a) and (b) distribution of two horizontally adjacent pixels in the plain and encrypted images presented in Figure 4. (c) and (d) distribution of two horizontally adjacent pixels in the same plain and encrypted images.

IV. Conclusion and Future Work

A novel approach based on chaos is presented in this paper for encrypting color images. The encryption/ decryption algorithms use 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 multi-point crossover and mutation operations, which result in image confusion and diffusion respectively. Several experiments are conducted and the results show that the proposed approach is capable of generating encrypted images with uniform distribution of the pixel values and very low correlation coefficients of adjacent pixels. It is also very sensitive to any changes in the secret key values. We are now working on modifying the proposed approach to handle each color component independently and to consider the inter-color correlation for increasing the secrecy of the cipher image.

Acknowledgment

The authors would like to acknowledge the support of King Fahd University of Petroleum and Minerals (KFUPM) and the University of Ha'il, Saudi Arabia, during development of this work.

References

[1] M. Padmaja and S. Shameem, "Secure Image Transmission over Wireless Channels," in Proc. of the Int. Conf. on Compu. Int. and Multimedia Applications, (ICCIMA 2007), 2007, pp.44-48.

[2] B. Furht, D. Socek, and A. M. Eskicioglu, "Fundamentals of multimedia encryption techniques," in B. Furht and D. Kirovski, (eds.) Multimedia Security Handbook, CRC Press, Ch. 3, 2005.

[3] X. Li and D. Zhao, "Optical color image encryption with redefined fractional Hartley transform," Int. J. for Light and Electron Optics, vol. 121, no. 7, April 2010, pp. 673-677.

[4] C.J. Tay, C. Quan, W. Chen, and Y. Fu, "Color image encryption based on interference and virtual optics," Optics & Laser Technology, vol. 42, no. 2, March 2010, pp. 409-415.

[5] K. Martin, R. Lukac, and K. N. Plataniotis, "Efficient encryption of wavelet-based coded color images," Pattern Recognition, vol. 38, no. 7, July 2005, pp. 1111-1115.

[6] W. Chen, C. Quan, and C.J. Tay, "Optical color image encryption based on Arnold transform and interference method," Optics Communications, vol. 282, no. 18, Sept. 2009, pp. 3680-3685.

[7] Z. Lin and H. Wang, "Image encryption based on chaos with PWL memristor in Chua's circuit," in Proc. of the Int. Conf. on Commun., Circuits and Systems, July 2009, pp. 964-968.

[8] C. Fu and Z. Zhu, "A chaotic image encryption scheme based on circular bit shift method," in Proc. of the 9th Int. Conf. for Young Computer Scientists, (ICYCS 2008), Nov. 2008, pp. 3057-3061.

[9] W. Yanling, "Image scrambling method based on chaotic sequences and mapping," in Proc. of the 1st Int. Workshop on Education Tech. and Computer Science, (ETCS '09), March 2009.

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

[11] G.M.B.S.S. Kumar and V. Chandrasekaran, "A novel image encryption scheme using Lorenz attractor," in Proc. of the 4th IEEE Conf. on Industrial Electronics and Applications, (ICIEA 2009), May, 2009, pp. 3662-3666.

[12] L. Luo, M. Du, B. He, F. Zhang and Y. Wang, "An image encryption algorithm based on spatiotemporal chaos," in Proc. of the 2nd Int. Congress on Image and Signal Proc., (CISP '09), Oct. 2009, pp. 1-5.

[13] C. Wei-Bin and Z. Xin, "Image encryption algorithm based on Henon chaotic system," in Proc. of the Int. Conf. on Image Analysis and Signal Proc., (IASP 2009), April 2009, pp. 94-97.

[14] D. Wang and Y. Zhang, "Image encryption algorithm based on s-boxes substitution and chaos random sequence," in Proc. of the Int. Conf. on Computer Modelling and Simulation, (ICCMS '09), Feb. 2009, pp. 110-113.

[15] N. J. Flores-Carmona, A. N. Pisarchik and M. Carpio-Valadez, "Encryption and decryption of images with chaotic map lattices," CHAOS Journal, vol. 16, pp. 1-6, 2006.

[16] Y. Honglei and W. Guang-Shou, "The compounded chaotic sequence research in image encryption algorithm," in Proc. of the WRI Global Congress on Intelligent Systems, vol. 3, May 2009, pp. 252-256.

[17] R. Matthews, "On the derivation of a chaotic encryption algorithm," Cryptologia, vol. 13, no. 1, Jan. 1989, pp. 29-41.

[18] Chong Fu, Zhen-chuan Zhang, Ying-yu Cao, "An improved image encryption algorithm based on chaotic maps," in Proc. of the 3rd Int. Conf. on Natural Computation, (ICNC 2007), 2007.

[19] N. Pareek, V. Patidar, K. Sud, Cryptography using multiple one-dimensional chaotic maps, Communications in Nonlinear Science and Numerical Simulation, vol. 10, no. 7, pp. 715–723, 2005.

[20] S. Liu, J. Sun, Z. Xu, "An Improved Image Encryption Algorithm based on Chaotic System", J. of Computers, Vol 4, No 11 (2009), 1091-1100, Nov. 2009.

[21] Y. Cao and Y. Fu, "Color image encryption based on hyper-chaos," in Proc. of the 2nd Int. Congress on Image and Signal Proc., (CISP'09), Oct. 2009.

[22] http://sipi.usc.edu/database/