Computer Science and Engineering 2012, 2(5): 77-85
DOI: 10.5923/j.computer.20120205.06

New Chaos-Based Image Encryption Scheme for RGB Components of Color Image

Nashwan A. Al-Romema[1,*], Abdulfatah S. Mashat[2], Ibrahim AlBidewi[2]

[1]Faculty of Computing and Information Technology- Rabigh
[2]Faculty of Computing and Information Technology- Jeddah, King Abdulaziz University, P.O. Box 344, Rabigh 21911, Saudi Arabia

Abstract The widely use of data digitization leads to the necessity of hiding the content of a message when it enters an insecure channel. The message prepared by the sender is encrypted before transmission. The encryption process requires an encryption algorithm and a key. In image cryptography, recently numbers of effective chaos-based image encryption schemes have been proposed. This paper presents a novel image encryption scheme which encrypts the pixels' RGB component instead of the pixels itself, where every pixel in the image will be extracted to its RGB triple, encrypted, and prepared to be transmitted. In the encryption process we use another image as a key, this image should be larger than or of the same size of the plain image that we intend to encrypt. The image encryption scheme introduced is based on chaotic systems. Simulation results show that the proposed image encryption scheme achieves high level of security, and reduces the encryption and the decryption time of image data.

Keywords Image RGB Color Components, Image Encryption, Chaos-Based Encryption, Chaotic Maps

1. Introductions

Hiding the content of data that are sent over an insecure channel has become one of the most important processes of some applications[1]. The message prepared by the sender is encrypted by using the encryption algorithm and key. In image cryptography, the focus has been putonsteganograph y, and in particular on watermarking during the last years[2]. Steganography is the science of hiding information and does not involve key[9]. The security of multimedia data in digital distribution networks is commonly provided by encr yption, which is the mathematical process that transforms a plaintext message into unintelligible ciphertext (confused). The classical image encryption such as Data Encryption Standard (DES) is designed with good confusion and diffusion properties[5], but it has the weakness of low-level efficiency when the image is large[10]. Classical and modern ciphers have all been developed for the simplest form of multimedia data. Chaos-based image encryption scheme is one of the encryption algorithms that have suggested a new and efficient way to deal with fast and highly secure image encryption[10]. Encrypting images, using chaos-based image encryption scheme, considers the image as 2-dimensional array of pixels[5]. Fridrich[8] suggested that a chaos-based image encryption processes go through two processes, i.e 1-chaotic confusion, and 2-pixel diffusion. In chaotic confusion process, the pixels of plain image are permuted with a 2 dimensional chaotic map, whereas the pixel diffusion alternates the value (gray-level) of each pixel in a sequential manner. Based on this architecture a number of chaos-based image encryption schemes have been proposed. Chen et al. employed a three-dimensional (3D) cat map[12] for image encryption. Guan et al.[11] used a two-dimensional (2D) cat map for pixel position permutation and a chaotic system that is called “The discretized Chen’s chaotic system” for pixel value masking. Wong et al.[5] proposed an idea to accelerate the encryption speed of the chaos-based image encryption schemes proposed by Lian et al.[7], in which the typical structure of chaos-based image encryption schemes is structured as the shown in figure 1. Each research in this area came up with a different algorithm of key generation and improvement of the chaotic encryption in terms of security and speed. Some algorithms proposed different ways of encryption by using hyper-chaotic to confuse the relationship between the plain-image and ciphered image[15]. All of these algorithm use one-dimension, two-dimension or three-dimension chaotic map with the advantage of high-level of efficiency. But their weaknesses materialize in the limitation of the key space used in chaotic system[1], and the complicated algorithm that includes large number of computer operations which might affects the performance of encryption/decryption processes. In addition to that the initial key values which might be easy for brute-force attack to predict. The generation of the key

* Corresponding author:
nashwan.alromema@gmail.com (Nashwan A. Al-Romema)

Published online at http://journal.sapub.org/computer

during encryption/decryption process, also affect the performance of the algorithm since you need to run different operations for this purpose. To overcome these drawbacks, we propose a new technique, which is very simple in implementation with high level of efficiency. The key is an image which is very difficult to be predict and the time for computation is reduced because no need to generate the key. The differences between our technique and the existing one are (1) We used the same methodology of chaos-based image encryption discussed in[5,7], with the change of the order of the stages to be performed in the image encryption/decryption, we run diffusion process and then confusion, this is because we use. (2) Number of iterations is one, as seen in figure 2 and can be compared with figure 1, where the encryption is performed in each pixel's RGB component rather than the pixel's value. (3) Our proposed technique is very simple in implementation. (4)The key is an image or matrix of random values, rather than generating key and add more complexity to the system. (5) the large key space is also one of the advantage of the proposed algorithm, since the key is an image which is very difficult to be predict, and the time for computation is reduced because no need to generate the key. (6) The technique can be extended to encrypt the video steams since the images in one frame are in the same size and one image-key is enough rather than generate key for each image in the frame. The rest of this paper is organized as follows. Section 2 presents the proposed algorithm. Section 3 describes the specification of the proposed algorithm. Section 4 describes the experimental result and Mean Square Error (MSE) results in varying some parameters of the algorithm, security analysis are given in Section 5. Finally, Section 6 concludes the paper.

2. Encryption Algorithm

The proposed image encryption algorithm is based on chaos-based image encryption scheme; it uses as a key another encrypted image or any matrix of random values larger or of the same size of the plain-image. The operations needed in encryption and decryption processes are reduced, with high level of security and less computational time. figure 2 shows the block diagram of the proposed chaotic encryption algorithm,

In algorithm 1, we explained the pseudocode of the proposed scheme

```
Algorithm 1: proposed chaos-based encryption scheme.
Input: (1) Plain image P with $M \times N$ size.
(2) Key image with $M \times N$ size.
Output: encrypted image with $M \times N$ size.
Begin
1. Get the plain image (P) with $M \times N$ size.
2. Get the secret key image (K) with $M \times N$ size.
3. For each pixel in P get RGB components ($P_r$,$P_g$,$P_b$).
4. For each pixel in K get RGB components ($K_r$,$K_g$,$K_b$).
5. for each corresponding RGB components in P & K apply diffusion process defined as
F(P(r,g,b) , K (r,g,b)) = C(r,g,b).
6. Concatenate RGB layers in C with binary representation for each (8 bit in each layer) in the order RGB.
7. Apply confusion processes (permutation with a selected 1D array of size 24 element.).
8. Get the permuted combination and divide it into three parts which will be the three RGB layers of C.
9. Repeat step 3-8 to all pixels in P and the corresponding pixels in K to get pixels in C.
End.
```

The encryption steps are as follows:

Step 1: Input to the algorithm the plan Image P with $M \times N$ size with key image K with $M \times N$ (same size).

Step 2: For each pixel of P get RGB components and for each pixel of K get RGB components.

Step 3: For each corresponding RGB components in P and K apply any logistic map to get new RGB components.

Step 4: Concatenate the new RGB component to be 24 bit.

Step 5: Perform the permutation process of the 24 bit with a selected 1D array of size 24 elements.

Step 6: Form the new permuted RGB component construct gray value for the new pixel.

Step 7: Perform steps 1 to 6 for all pixels in P.

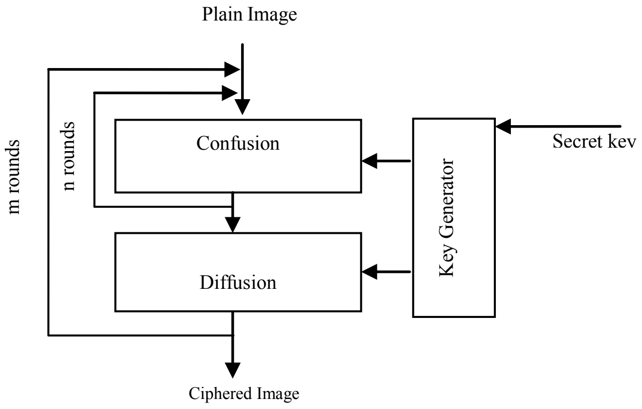


Figure 1. Typical architecture of chaos-based image encryption scheme

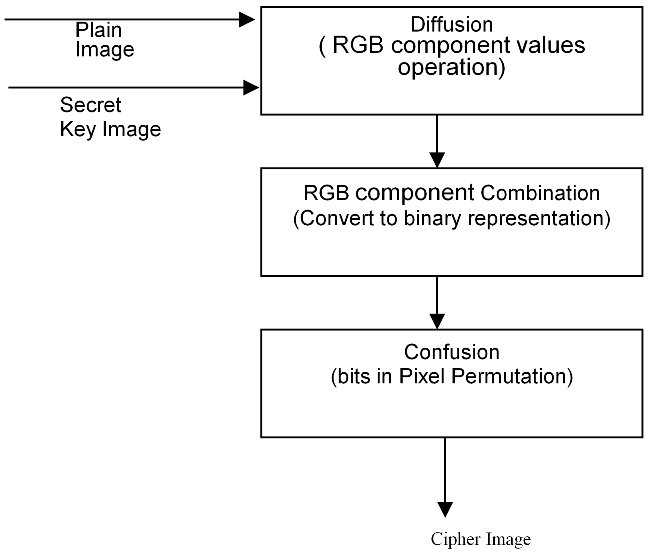


Figure 2. Typical architecture of the proposed image cryptosystems

3. Scheme Specification

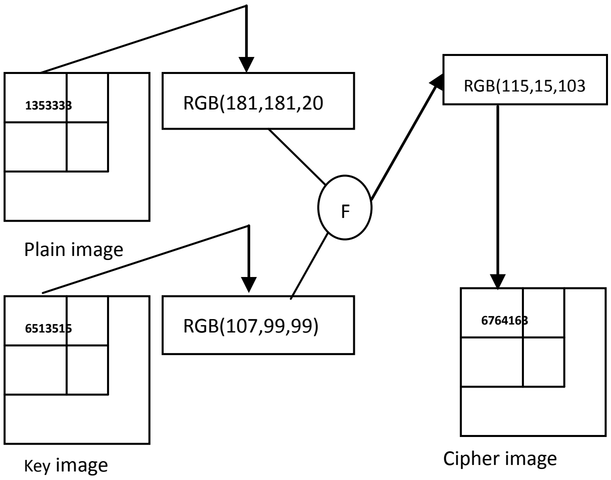


Figure 3. Example of the proposed image encryption scheme

As in algorithm 1, the process of the proposed algorithm goes through different steps as shown in figure 3, several issues should be noted, the first one is the chosen key should be larger or of the same size as the image we indented to encrypt. In case of the chosen key does not have the same dimension of the plain-image, the system will encrypt the pixels of the plain-image with the corresponding pixels in the key-image. The pixels in the plain-image without any corresponding pixels in the key-image will not be fully encrypted as can be seen in Figure 3. This is why the dimension of the key-image should be the same or larger than the dimension of the plain-image. The second factor that might affect the efficiency and security level of the scheme is the diffusion processes which can be defined as any arithmetic operation/function performed on the pixel's RGB components. Such function should be strictly monotonically function[3], it means that the encrypted pixel can be recovered back in the decryption process, or this will lead to increase the MSE in the decrypted image. High MSE means that data does not recovered back in decryption process (we lost data), which is not desired image encryption schemes.

In contrast to the chaos-based scheme discussed in[5,7], our scheme has the following differences: first the order of the stages to be performed in the image is different, we perform diffusion and then confusion. Secondly, one iteration in sequential fashion is performed in each pixel's RGB component rather than the pixel's value. The two stages process as the following:

Stage 1: Diffusion operation, every pixel in plain image P, and the corresponding pixel in the key image K are extracted in to their RGB components.

$$\text{Extract }(P) \Rightarrow P_r, P_g, P_b$$

$$\text{Extract }(k) \Rightarrow K_r, K_g, K_b$$

Then, simple operation Θ is performed in each two corresponding pixel's RGB component to get the corresponding RGB component in the cipher image C.

$$C_r = P_r \,\Theta\, K_r$$

$$C_g = P_g \,\Theta\, K_g,$$

$$C_b = P_b \,\Theta\, K_b\,,$$

The nature of Θ operation/function depends on the security level we want to achieve; this function can be linear, nonlinear or geometric function. The Θ function should be strictly monotonically function. Computation time is a function of the complexity of the function and number of operation required to be performed in this stage. In our proposed scheme we assumed Θ = XOR and we get the result shown in experimental result section. We finish stage1 for each pixel by getting corresponding pixel's RGB components in C, the output of this stage is (C_r ,C_g ,C_b).

Stage 2: The output of stage1 is considered as the input to this stage. The operation we perform in this stage is the binary function, which converts each RGB component into binary, then concatenate these results as follows:

Bin((C_r,C_g,C_b)) ===>[011010000100000001010101] 24 bit

The pixel's RGB values are in the range from 0 to 255, so each RGB component occupies 8-bits. The result of concatenate((C_r,C_g,C_b)) will be 24-bits. The last operation in this stage is to perform permutation in the 24-bits on one dimension array of binary data; this array is of size 24-bits. The goal of this operation is to get 24 permuted bits that will be divided into three components (8-bits each). The purpose of this permutation used is to confuse the relationship between the plain image and cipher image. We get the integer value corresponding to each block of 8-bits to be the final RGB encrypted values. These values will be combined to get the encrypted pixel in the cipher image.

4. Experimental Analysis

Several images have been encrypted by using the proposed chaotic image encryption scheme. To measure the error in the plain-image and the encrypted -image, we use MSE error metrics[16]. The mathematical formula for MSE is as follows:-

$$\mathrm{MSE}=\frac{\sum\limits_{i=1,j=1}^{M\quad N}\sum[(\mathrm{p}(\mathrm{i},\mathrm{j})-\mathrm{c}(\mathrm{i},\mathrm{j}))]^2}{M\times N} \qquad (1)$$

Where p(i,j) is the original image, c(i,j) is the encrypted image and $M \times N$ are the dimensions of the P and C images. A lower value for MSE means lesser error. We measure the MSE for each RGB components in plain-image and RGB components in ciphered-image. Example: if we have an image P of size $L_P= M \times N$ and the encrypted image C of size $L_C = M \times N$, then the MSE in our study can be calculated for each R, G, and B color components in the plain-image with the corresponding R, G, and B color components in ciphered-image as the following:-

We assume

1. R_P, is the Red color component in Plain-image P, R_C, is the Red color component in Ciphered-image C.

2. G_P, is the Green color component in Plain-image P, G_C, is the Green color component in Ciphered-image C.

3. B_P, is the Blue color component in Plain-image P, B_C, is the Blue color component in Ciphered-image C.

$$\mathrm{MSE}_R=\frac{\sum\limits_{i=1,j=1}^{M\quad N}\sum[(R_P(\mathrm{i},\mathrm{j})-R_c(\mathrm{i},\mathrm{j}))]^2}{M\times N} \qquad (2)$$

$$\mathrm{MSE}_G=\frac{\sum\limits_{i=1,j=1}^{M\quad N}\sum[(G_P(\mathrm{i},\mathrm{j})-G_c(\mathrm{i},\mathrm{j}))]^2}{M\times N} \qquad (3)$$

$$\mathrm{MSE}_B=\frac{\sum\limits_{i=1,j=1}^{M\quad N}\sum[(B_P(\mathrm{i},\mathrm{j})-B_c(\mathrm{i},\mathrm{j}))]^2}{M\times N} \qquad (4)$$

MSE can be expressed as MSE = (MSE_R, MSE_G, MSE_B) which is the error in the three color component. The factor that increases the MSE is the wrong chosen key, either in size or structure. Another factor that increases the MSE in our proposed encryption algorithm is the function in the diffusion process, that does not satisfy strictly monotonically function. figure 4 shows the experimental result for encrypting an image of size 200 × 312 with a key image of the same size. The image in figure 4(a) is the original image and the image in the figure 4(b) is the key image. The encrypted image is shown in figure 4(c).

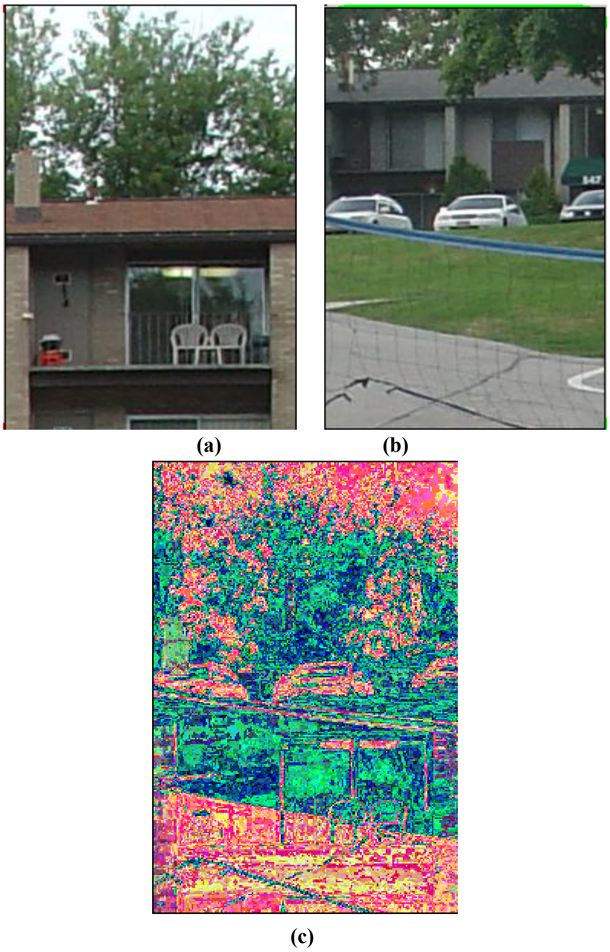
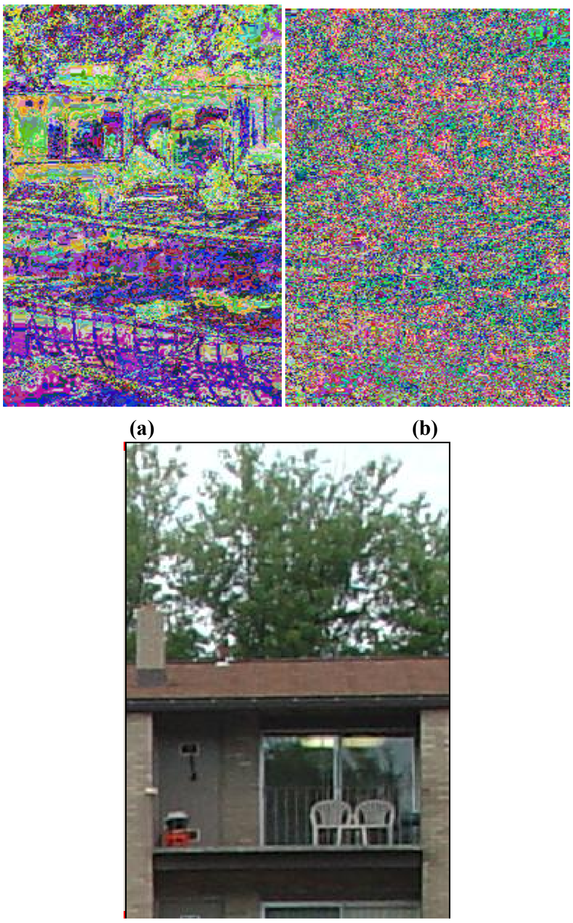


Figure 4. Image encryption experimental result 1: (a) Plain- image, (b) Key image, (c) Encrypted image

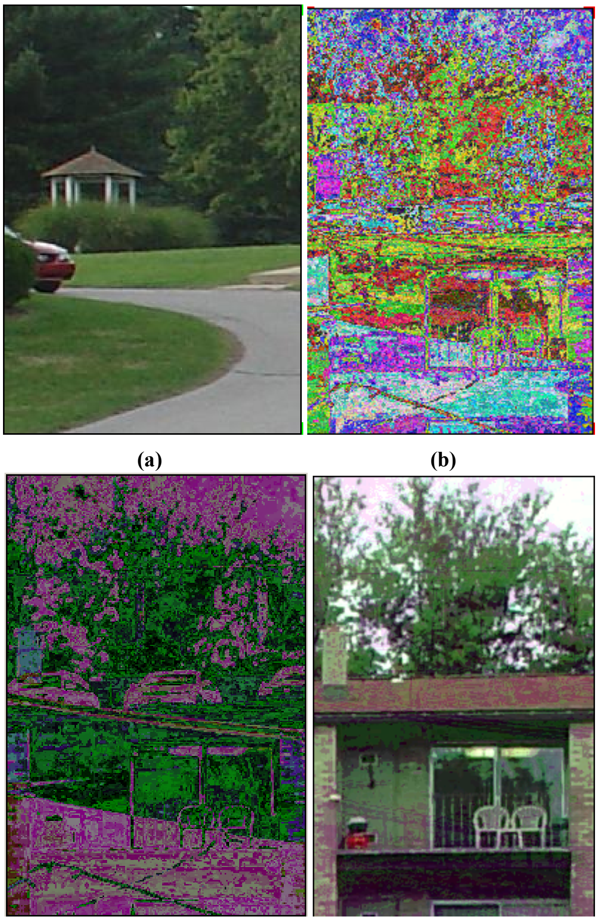
As we can see in figure 4(c), the encrypted image still has some details that are not desired, encrypting the key image first and encrypt the original image with the encrypted key image is the best solution we have encountered in our research. The key image in figure 4(b) is first encrypted by itself, which means, it is the original image and the key image. After getting the encrypted key image as shown in figure 5(a), then the original image in figure 4(a) will be encrypted using this key image to output the encrypted image as shown in figure 5(b). As can be seen in figure 5(b) the encrypted image is unknowable. In the decryption processes, if we use the same key that has been used in encryption, we get the same image as the original image, it can be seen in figure 5(c) that the decrypted image is clear and correct without any distortion of the image. The MSE = (0, 0, 0) which means no differences between the original image's RGB components and the decrypted image's RGB components. Another case, if the decryption processes use different key from that one used in encryption process like the key shown in figure 5(a), the decrypted image will not be as the original image; more noise will be added to the image as shown in figure 5(b). The MSE is in this experiment increased which is roughly (9978.6, 9834, 10596) to indicate the difficult recognized image details.


(a) (b)

(c)

Figure 5. Image encryption experimental result 2 using encrypted key image: (a) Encrypted key image, (b) The encrypted image using the encrypted key image, (c) The decrypted image using the correct key

We can conclude from this, that the proposed chaotic encryption scheme is sensitive to the key and small change to the key or using different key will not get the correct plain-image. One more thing should be mentioned here is the function which we us in diffusion processes. If this function is non-strictly monotonically function, then the encryption will be done smoothly, like the encrypted image in figure 5(c). But the problem in this case is that, when we apply the decryption processes even though we use the same key that is used in encryption process, the error will be increased and the decrypted image is not the same as the plain image. The decrypted image will not be as the original image, some noise will be found in decrypted image as shown in figure 5(d). The MSE is increased which is roughly (1087.4, 1544.5, 1091.6) to indicate noisy image.


(a) (b)

(c) (d)

Figure 5. Image encryption experimental result 3 using encrypted key image: (a) Wrong key image, (b) Decrypted image using wrong key, (c) Encrypted image using non strictly monotonically function in diffusion stage, (d) The decrypted image using non strictly monotonically function

Another experiment has been done on Lena image as shown in figure 7; figure 7(a) is Lena plain-image of size 225 × 227. figure 7(b) is the image- key of the same size 225 × 227. figure 7(c) is the encrypted image. figure 7(d) is the decrypted image with the right key begins used in encryption process.

5. Security Analysis

The image encryption scheme presented in this paper is based on chaos-based image encryption scheme suggested by Fridrich[8]. Good encryption procedure should be robust against all kinds of known attacks[6]. The encryption procedure should be sensitive to the secret key and using different keys should not recover the plain-image. In our algorithm using different key will add more noises to the encrypted image instead of getting the plain-image as shown figure 5 experimental result 3. In our proposed algorithm, the key is a selected image which is larger or of the same size of the plain-image. using this key in encryption /decryption processes will overcome the drawbacks of the limited key space in[10] and satisfy high level of security. This is because it is very difficult to predict matrix of values, whereas using initial values for key generation might be feasible for brute force attack. Comparing the proposed technique with the latest existing technique proposed by M. Francois et al [17], in terms of security and complexity we found that our technique satisfied high level of security. In figure 6 we can see the Fairy's image after encrypted by M. Francois's algorithm and the same image after encrypted by our algorithm. It can be seen that we have satisfied high level of security as our encrypted image looks like the one in [17]. In addition to that our algorithm is simple in implementation as number of operations is less than the number of operations in Francois et al [17]. Our algorithm is more secure as the key space is very large and it's difficult to be predicted.

5.1. Distribution of Pixels

The proposed image encryption algorithm experimentally tested on "Lena" image 225×227, and because our algorithm works on RGB images which are *M×N×3*, where the image can be considered as 3 dimensional image, **s**o we need to get from *M×N×3* image 3 different 2 dimensional images of size *M×N* for R, G, and B components of the image, because histogram can only applies on 2 dimensional image. We use this functions *rgb2gray* in matlab to convert RGB image into grayscale images. figure 8(a) shows the histogram of Lena image after converted form RGB image into grayscale image and the histogram of the encrypted image after converted form RGB into grayscale, but for more accurate results we need to get the histogram of R component of Lena image, and the histogram of R component of Lena encrypted image as shown in figure 8(c) and figure 8(d), the same we do on G and B component of Lena image and Lena encrypted image as shown in figure 8(e), figure 8(f), figure 8(g), and figure 8(h). As a result of this, we can see in figure 8 the pixels distributions of the original image and the pixels distributions of the encrypted image. Encryption process returns noisy images. The histogram of encrypted images in figure 8(d), figure 8(f), and figure 8(h) are very close to uniform distribution which are different from that of the corresponding original images in figure 8(c), figure 8(e), and figure 8(g). These aspects indicate high level of security against the known-plaintext attack[14].

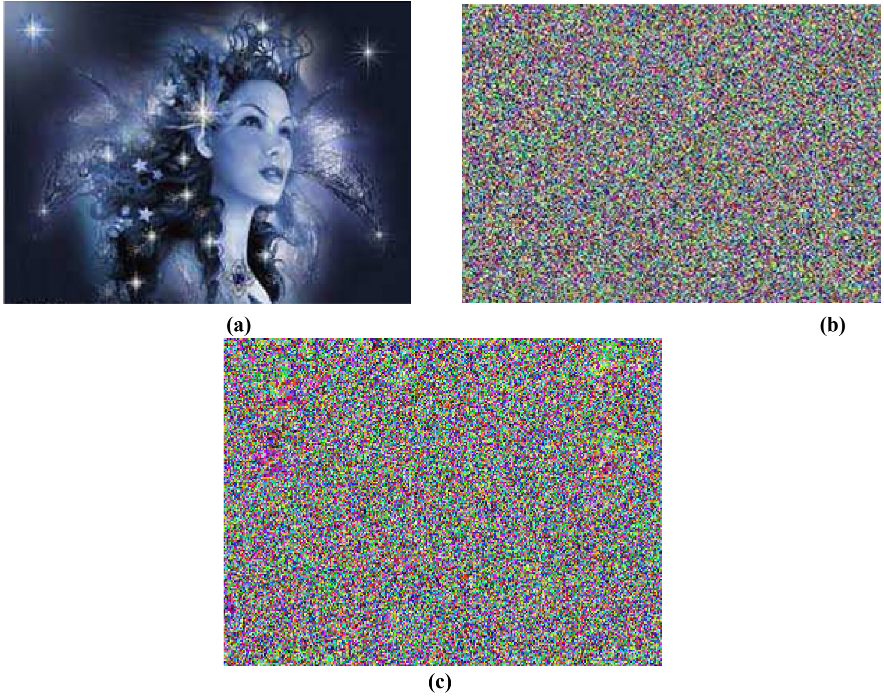
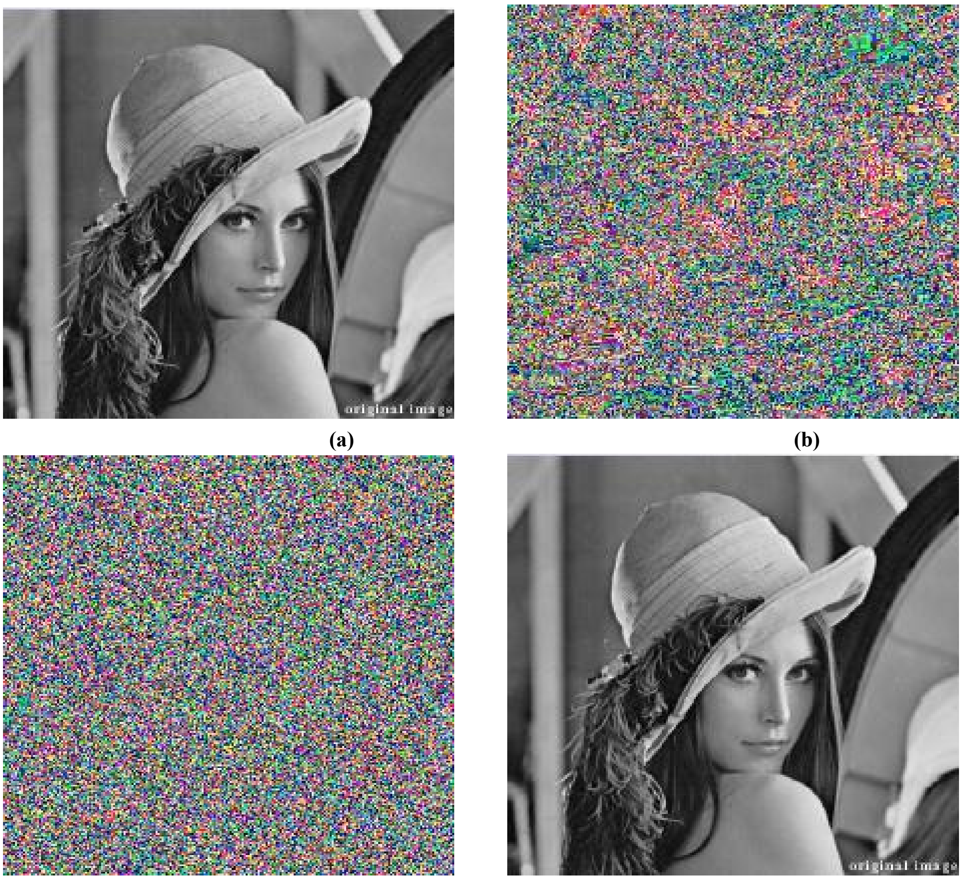


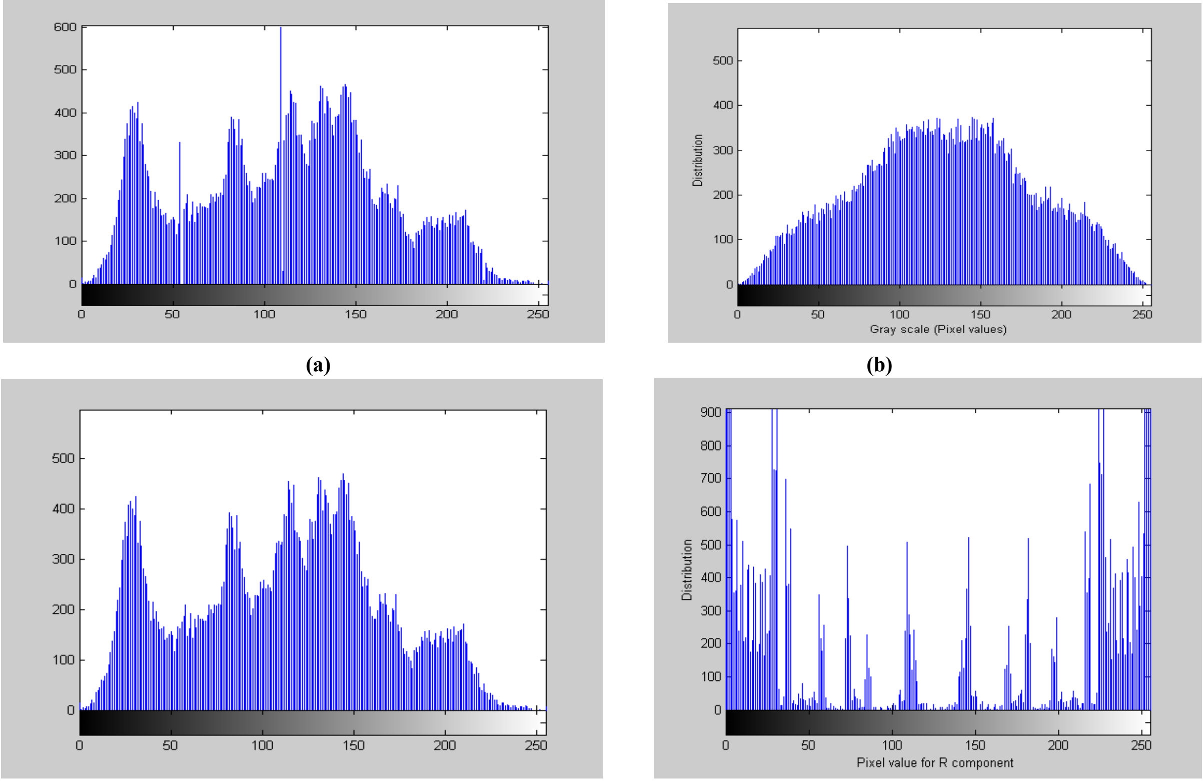
Figure 6. Comparison of our image encryption technique with M. Francois's technique [17] : (a) Fairy's Plain- image, (b) Fairy's encrypted image by using Francois et al [17], (c) Fairy's encrypted image by using our algorithm



(a) (b)

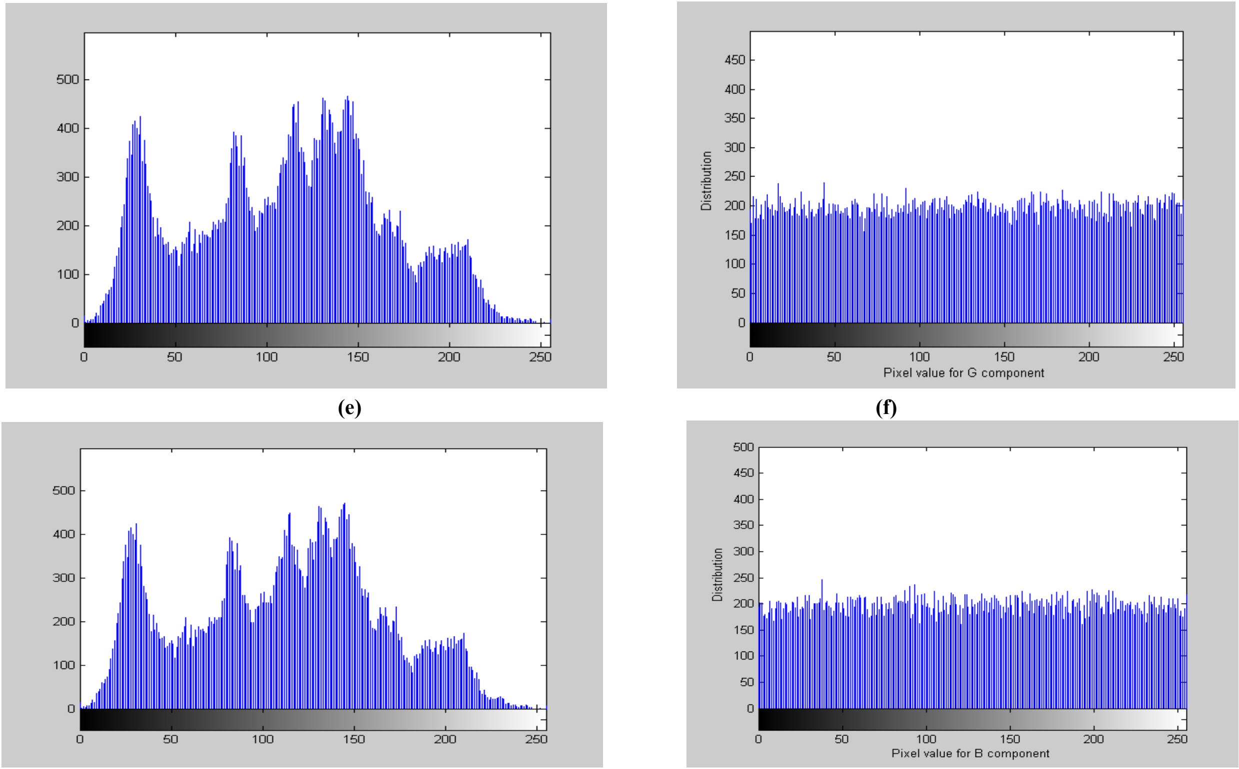
(c) (d)

Figure 7. Image encryption experimental result 4 using Lena Image: (a) Plain- image, (b) Key image, (c) Encrypted image, (d) The decrypted image



(a) (b)

(c) (d)



(e) **(f)**

(g) **(h)**

Figure 8. Histogram of plain-image and encrypted image (a) Lena Plain-image histogram after converted from RGB scale into grayscale, (b) Histogram of Lena Encrypted image, (c) Histogram of Lena's image Red color component, (d) Histogram of encrypted Lena's image Red color component, (e) Histogram of Lena's image Green color component, (f) Histogram of encrypted Lena's image Green color component.(g) Histogram of Lena's image Blue color component, (h) Histogram of encrypted Lena's image Blue color component

6. Conclusions

In this paper, we introduced a new image encryption algorithm based on chaos- encryption scheme. The encryption process is applied on RGB components of the image's pixel instead of the pixels itself. The encryption process uses another image as a key, this key-image should be larger or of the same size of the plain-image. In encryption process the corresponding RGB components in plain-image and key-image go through diffusion process and after that the ciphered image's RGB components are permuted on 1 dimensional array. To overcome the limitation of key space in[13] and the improved NCA algorithm in[10], we proposed our key to be an image or a matrix larger than or of the same size of the plain image. Using the key in this way, we reduced the number of operations for key generation as well as the key is secret to make brute-force attacks infeasible. Several Experiments were carried out with numerical analysis, to demonstrate the high level of security in the proposed image encryption scheme. The prototype system of the proposed algorithm can be used in any application of image transmission over unsecured channel. In the future work we intend to make our system flexible in key selection, so the system can generate the key as a matrix of random values larger or of the same size as the images we intend to encrypt.

REFERENCES

[1] Ismail, I., Amin M., and Diab H.,"A Digital Image Encryption Algorithm Based A Composition of Two Chaotic Logistic Maps",International Journal of Network Security, Vol.11, No.1, PP.1–10, July 2010.

[2] Droogenbroeck, M., and Montefiore, R., Techniques for a selective encryption of uncompressed and compressed images, In ACIVS Advanced Concepts for Intelligent Vision Systems, Ghent, Belgium, pages 90-97, September 2002.

[3] Gonzalez, R., woods R., Digital Image processing, Prentice Hall; 3rd edition, 2007, ISBN-13: 978-0131687288.

[4] Fridrich J., Symmetric Ciphers Based on Two-dimensional Chaotic Maps". Int. J.Bifurcat Chaos 1998;8(6):1259-84.

[5] Wong k., Kwok B., and Law W., A Fast Image Encryption Scheme based on Chaotic Standard Map, CoRR abs/cs/0609158: (2006), City University of Hong Kong, arXiv:cs/0609158v1[cs.CR], http://arxiv.org/abs/cs/0609158.

[6] Ahmed H, Kalash H, and Farag Allah O., An Efficient Chaos-Based Feedback Stream Cipher (ECBFSC) for Image Encryption and Decryption, Informatica 31 (2007) 121–129.

[7] Lian, S., Sun, J., Wang, Z., Security analysis of a chaos-based image encryption algorithm, Physica A 2005; 351:645-61.

[8] Kocarev L., Chaos-Based Cryptography: A Brief Overview, 1531-636X/10/$10.00©2001IEEE

[9] Ariffin M., Chaos Based Cryptography an Alternative to Algebraic Cryptography, VOLUME 2 NUMBER 1 (2008).

[10] Gao H., Zhang Y., Liang S., Li D., A new chaotic algorithm for image encryption, science direct, Elsevier Ltd, chaos, solutions and fractals 29 (2006) 393-399

[11] Guan ZH, Huang FJ, Guan WJ. Chaos-based image encryption algorithm, Phys Lett, A 2005;346:153-7.

[12] Chen G, Mao YB, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons & Fractals 2004;12:749-761.

[13] Sobhy MI, shehata AR. Method of attacking chaotic encryption and countermeasures. IEEE Acoust Speech Signal process 2001:1001-4.

[14] Giesl, J., Vlcek, K., Image Encryption based on strange attractor. ICGST-GVIP Journal, ISSN 1687-398X, Volume (9), Issue(II), April 2009

[15] Gao, T., Chen, Z. A new image encryption algorithm based on hyper-chaos, science direct, 2007.

[16] Z. Wang and A. C. Bovik, “Mean Squared Error: Love it or Leave it?—A New Look at Fidelity Measures,” IEEE Signal Process. Mag., vol. 26, no. 1, pp. 98–117, 2009.

[17] M. Francois, T.Grosges, D.Barchiesi, R.Erra, A new image encryption scheme based on a chaotic function, 2012, Signal Processing: Image Communication 27 (2012)249–259