ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 9(8), DOI: 10.17485/ijst/2016/v9i8/79067, February 2016*

Text Embedded into Encrypted Image based on Genetic Algorithm on Piece-Wise Linear Chaotic Map

Bano Mahwish[1], T. M. Shah[2] and Shah Tariq[1]

[1]Department of Mathematics, Quaid-e-Azam University, Islamabad, Pakistan.
[2]Department of Mathematics, Air University, Islamabad, Pakistan. Maham_dia@yahoo.com, dr.tasneem@mail.au.edu.pk, stariqshah@gmail.com

Abstract

Objective: To make texts and images secured by placing into. **Method:** Use of genetic algorithm for the selection of pixel's position in the plain images in which a class of fitness was used based on the intensity of the pixels influenced by neighbouring pixels. The algorithm presented in our previous work was secured and useful for text embedding where intensity of each pixel has an important role both in grey and colour images. **Findings:** Algorithms for text embedding into images have developed by a number of researchers. They simply placed the text into images sequentially and randomly. This has made text secured and difficult to retrieve but might not be impossible to retrieve. In the present paper, we proposed an added algorithm to Genetic Algorithm on Piece-Wise Linear Chaotic Map (PWLCM) for text embedding. This could have a double security on text encryption into images where images are itself encrypted. This made the diffusion processes more secured. Results and tests analysis are presented for comparison with existing text embedded algorithms in the literature. **Applications:** Public communication network, medical images are protected from forging, secret data and documents are to be safe from intruders.

Keywords: Encrypted Image, Genetic Algorithm, Pixel's Intensity, PWLCM, Text Embedding

1. Introduction

The users of network communication of 3G and 4G technologies, the digital images are commonly transmitted in public communication network, therefore, it is important to protect images from piracy, most importantly, the medical images must be protected from forging, so the image encryption technology has become a powerful tool for cryptography. The images are of two types: The heavy data capacity pixels and strong correlation with the neighboring pixels. Therefore, the conventional encryption algorithms do not suitable for such type of image encryption. Recently[1,2,6-9] researchers have intensively used chaotic cryptography as it has ergodicity and sensitive dependence on initial conditions, high efficiency in image encryption. There are many rooms to improve the strength of the algorithm such as optimal randomness in placing the pixels or acquiring genetic algorithmic properties for the optimally placement of pixels in the process of confusion and diffusion in cryptography.

The confusion and diffusion processes based on Shannon theory[10] are applied in image encryption successfully. These processes have used a permutation diffusion structure and other were used chaotic image encryption systems, both of them shuffle the position of the pixels by confuse phase, and separated the two processes.

These are weak algorithms and periodic state appeared after little iteration[11]. Other problems are the size of a picture must be the same that is, height and width of the processed images must be same for permutation of processes and the pixel position of different images in same size is fixed. These weaknesses are not acceptable in the practical application in chaotic cryptography.

**Author for correspondence*

Separation of confusion and diffusion processes reduced the efficiency of encryption algorithm. Earlier researchers[12] were confused the plain image by using a different map scheme and then diffused with a piece-wise linear chaotic map. In[5], we developed an improve PWLCM model based on random selection of pixels and the pixels fitness of plain images. The fitness may be selected by a simple use of genetic algorithm. We used this model to shuffle positions and diffuse values of pixels in plain image simultaneously. Test results and security analysis were carried out and the strength of the algorithm achieved to resist against brute force attack. The present paper is an addition to our earlier work[5] by introducing technique to embed text into the encrypted images. The researchers[4,5] have implemented this idea of embedding text into image without taking or analyzing the strength of security. The only security was the hiding text into images. This could only be secured till the retrieval of the images. We proposed an idea that first encrypt the text and image both using conventional text encryption technique and encrypt the image using the genetic algorithm. Then randomly place the encrypted text into encrypted images. This is called the double encryption techniques. Results are obtained and carried out time analysis for the retrieval of text. Further this technique can be used into medical images where the patient's history could also be placed into their encrypted bio-logical images.

2. Algorithm

The Modified-PWLCM (MPWLCM)

The PWLCM can be described as:

$$x_{n+1} = \begin{cases} x_n / q & \text{for } 0 \le x_n < q \\ (x_n - q)/(0.5 - q) & \text{for } q \le x_n < 0.5 \\ F(1 - x_n, q) & \text{for } 0.5 \le x_n < 1 \end{cases}$$

Where $x_N \in (0, 1)$, and the control parameter which is so called secret key, $q \in (0, 0.5)$, evolves into a chaotic state. This system provides excellent profile of and due to its randomness; it is widely used for image encryption. An improved version based on this PWLCM was proposed in[4] given below as:

$$x_{n+1} = \frac{x_n - \left\lfloor \frac{x_n}{q} \right\rfloor \times q}{q} \qquad (1)$$

Figure (3) and Figure (4) show the state sequences of PWLCM and MPWLCM respectively. From the figures it is shown that the randomness of MPWLCM is better for the image encryption. In the following section we proposed a crypto system based on these two but selection of pixels would be done by genetic algorithm.

Genetic Algorithm for MPWLCM

Based on PWLCM and MPWLCM, we calculated the sequence x_{N+1}, converted this into binary sequence s_{N+1} and applied genetic algorithm to find the most fit pixels, s'_{N+1}. Fitness of the binary sequence is based on the intensity of each pixel compared with the 8 neighboring pixels. Following are the steps of confusion, diffusion and optimization of the plain image pixels:

- Calculate x_{N+1} from PWLCM algorithm
- Re-calculate x_{N+1} using MPWLCM
- Convert x_{N+1} into binary sequence, s_{N+1}
- Apply genetic algorithm to calculate the fitness of the algorithm based on intensity of each pixel compared with 8 neighboring pixels and obtained the sequence s'_{N+1}
- Convert s'_{N+1} back into decimal values as y_{N+1}.

The GA-MPWLCM is better performed in randomness and highly secured system information encryption.

The Cryptosystem

Consider a gray scale image of size L = M × N, where M, N are rows and column matrix and values of each element are ranged from 0 to 255. The data treated as one-dimensional array $p = (p_1, p_2, \ldots, p_L)$, where p denotes the gray level of the image pixel at

$\lceil(i/N)\rceil$ row and $[\lceil(i/N)\rceil - (i - 1) \times N]$ column.

Generating Permutation Sequence

For a given value x_0 and p, we generated sequence to change the position of image pixel. Calculate the one-dimensional array T (i), as $T = [t_1, t_2, \ldots, t_L]$, an ergodic matrix of size 1 × L, where t(i) are integers, $t(i) \in [1, L]$, and $t(i) \neq t(j)$ if $i \neq j$. The scheme is as follows:

Iterate the PWLCM as $x_{i+1} = F(x_i)$ by using equation (1) for n-times to make it steady state, n, is a constant; set a one-dimensional matrix AAA, with zero elements of length L; initialize the permutation sequence T = [t(1), t(2), ..., t(L)] : T = AAA

- Let $i = 1$;
- Iterate PWLCM to calculate a new x, compute an integer j using
- current x as follows:
- $j = |(\lfloor(x \times 10^{15})\rfloor, L)| + 1$
- Checking values of j and AAA(j), if (j == i), or(AAA(j) == 1) then repeat the previous step; else go to next step.
- AAA(j) = 1; t(i) = j
- Increase i by one and repeat the whole process until i reaches L.

Generating Diffusion Sequence

Steps to generate the diffusion sequence are as follows:
The diffusion sequence is denoted by $K = [k_1, k_2, ..., k_L]$, $k(i) = 0, i = 1, 2, ..., L$

- Let $i = 1$
- Compute a new x, by iterating PWLCM using the current x as follows:
- $k(i) = |(\lfloor(x \times 10^2 - \lfloor(x \times 10^2)\rfloor) \times 10^3\rfloor, 256)|$.
- Checking for k(i), if $k(i) < 3$, then $k(i) = k(i) + 3$
- Increase i by one and repeat the whole process until i reaches to L.

Encryption Algorithm

The encryption process uses both permutation sequence T to shuffle the position of image pixels and uses diffusion sequence K to defuse the values of image pixels simultaneously. The encryption algorithm is described as follows:

- Move the pixel position i in plain image to the position j in the cipher image, where $j = t(i)$.
- Simultaneously the pixel value of position i in plain image is altered by using diffusion key k(i) and previously encrypted pixel value.
- The swapping of pixel values position is done randomly by selecting two positions of plain image and encrypted image. This algorithm is different from the usual algorithms and gave more strength to the security.
- The permutation and diffusion processes may repeat R rounds ($r = 1$toR, $R \geq 1$).
- The encryption algorithm is the same as given in [3], except the selection of plain and cypher pixels is done randomly as follows;

$$c(j) = |(p(i) + c_1, 256)| \oplus k(i) \text{ if } r = 1$$
$$c(j) = |(c(i) + c_1, 256)| \oplus k(i) \text{ if } r > 1$$

Where $i = 1, 2, \ldots, L$ and $j = 1, 2, \ldots, L$
The encryption algorithm is described as follows:

- Set $n = 1$
- If n is even, then $i = \lceil(n/2)\rceil$; otherwise $i = (RanL - n/2 + 1)$
- Obtain $j = t(i)$
- Use the above mentioned formulas to permute and diffuse the current pixel simultaneously.
- Upgrade $n = n + 1$
- If $n < L$ then repeat the whole steps, otherwise one round is completed.

Embedding Phase

To insert the text into the least significant bits of image, the algorithm is as follows:
Inputs: An image in which secret text is to be embedded.
Output: An image in which text is embedded

Procedure:

Step 1: Take out all the pixels in the target image and accumulate it in the array which is called pixel array.

Step 2: Take all the characters from the secret text and put it in the array which is called character array.

Step 3: Take all the Stego-key characters and put it in the array, called Key array.

Step 4: Select first pixel location and take a character from keyarray and put it in the first section of pixel, if keyarray contains more character then put remaining characters in the first section of upcoming pixel, or else follow step (e), e has been utilize as an end mark.

Step 5: Put a few ending symbol which point out closing tags of the key. In this algorithm '0' has been utilized as an ending mark.

Step 6: The component of character array which are characters, substitute these characters in every first section of upcoming next pixels.

Step 7: Reiterate the previous step, unless all the characters have been replaced with each pixel of the target image.

Step 8: Here must be given any mark which point out that data has finished.

Step 9: Output, An image in which all the secret text character has embedded.

Extraction Phase

To retrieve the secret message or text; follow the procedure given below:

Inputs: Steganographed image

Result: The text which is embedded by sender into the image

Process:

Step 1: Take character array, key array, and pixel array.

Step 2: Take out all pixels from the steganographed image and put it in the pixel array.

Step 3: Examine the pixels and take out key characters from first section of pixel and put it in the key array. Repeat this step, till the ending mark, or else go next step.

Step 4: Now, after saving the key array, if the retrieved key is equal or similar with the sender key then take step 5, or else stop the process by showing warning that key is not recognized or got similar.

Step 5: After validation of the key, again examine the next pixel and take out embedded text character from the first section of upcoming pixel and put it in the character array. Do this Step until the ending mark or else go to further step

Step 6: Take out all embedded text which is stored in character array.

The most important task of this algorithm is to enhance the PSNR and utilize the logic gates to achieve this goal.

Step 1: Take the target or secret image

Step 2: Take the cover or wrap image

Step 3: Put the number of significant bits i.e. n; where n=1,2,3,4,5,6,7,8

Step 4: Set the size_secret image= size (secret) and size_cover image= size (Cover Image)

Step 5: Put the number of significant bits "n" significant bits of every byte of wrap image to zero by utilizing bit through AND logic gate process on wrap image and size_secret image matrix

Step 6: Fixed or inserted the MSB of target image to make steganographed image by utilizing Steganographed_img= (coverzero_secret)/2^{8-n}

Step 7: Retrieve the coded or secret image

Step 8: Show steganographed image and embedded image

Step 9: End

Reminder: When the value of n is increased the worth of both images would be decreased.

This algorithm is applicable for 24 bit and 8 bit grayscale images.

Experimental Result:

After the experiment, outcomes have shown the strength of this algorithm as compare to the others[5]. We embedded and hide the text in actual or targeted image and got steganographed image. The peak signal to noise ratio (PSNR) of steganographed image is analyzed, the PSNR increased in this algorithm and visually, one cannot differentiate between original image or wrap image and steganographed image. This algorithm is implemented in Netbeans Java. See Figures (1-2).

3 Results and Security Analysis

Experiment was performed using 256×256 images with 8-bit gray scale, with parameters, $q = 0.3$, $x_0 = 0.27$, nmax = 200, $c_0 = 150$, and $R = 2$, results were presented in Figures (3-12). The encryption algorithm has large enough key space to resist all kind of brute force attacks.

Key Space Analysis

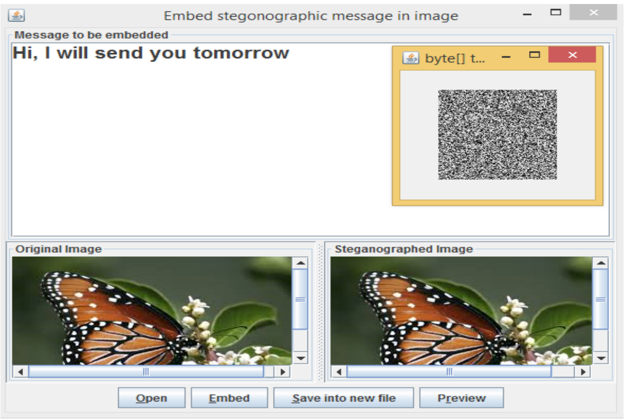
In this paper, the images for testing are taken as 256×256 with 8 bit grayscale. We assumed the same system parameters as in the case of MPWLCM given in[3]. Key space size is generally the total number of keys are used for encryption processes. Since we are using the MPWLCM scheme, therefore it is guaranteed that the total size of the key is more than 2^{124} due to the genetic algorithm. So the present algorithm has a large enough key space to resist all kinds of attacks.

Statistical Analysis

Generally the confusion and diffusion processes are known statistical analysis. This analysis has been done for MPWLCM and for the genetic algorithm also. These are shown on histograms (Figures 7,8,11,12) and the correlations of adjacent pixel in encrypted images in Table (2).

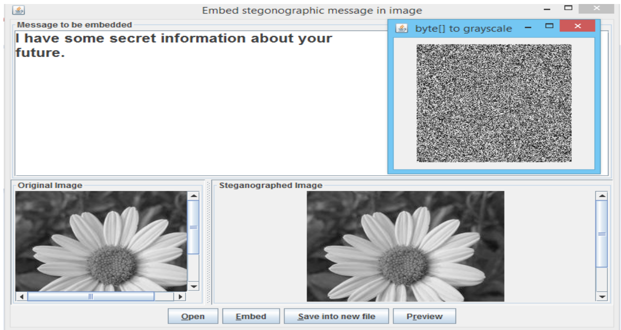
Histograms of Encrypted Images

To demonstrate the histogram, we have selected images of 256 gray level images with size of 256×256 and calculated their histograms. The histogram of encrypted image



Hi, I will send you tomorrow

Figure 1(a). "Hi, I will send you tomorrow" Message is going to embed into an image (b) Selected Butterfly Image (c) After Embedding Steganographed Image (d) Encrypted Image.



I have some secret information about your future.

Figure 2(a). "I have some secret information about future" Message is going to embed into an image (b) Selected flower Image (c) After Embedding Steganographed Image (d) Encrypted Image.

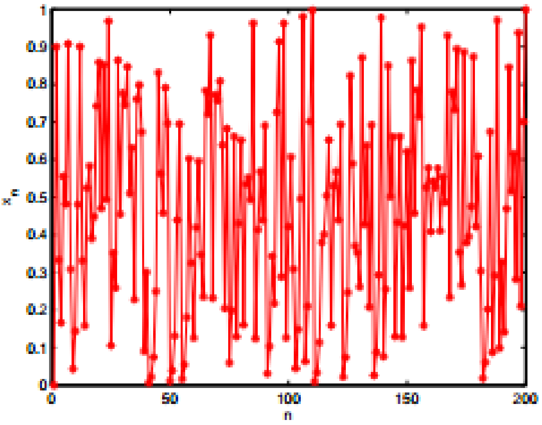


Figure 3. The state sequences of MPWLCM (q=0.3).

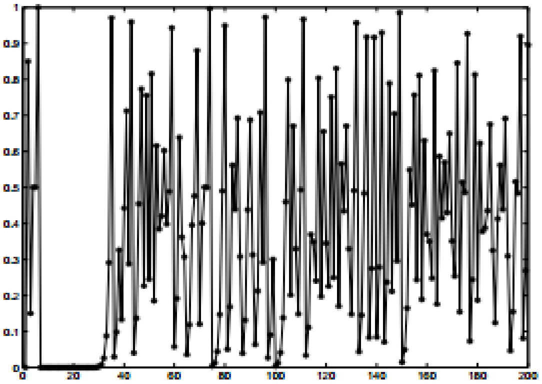


Figure 4. The state sequences of PWLCM (q=0.3).

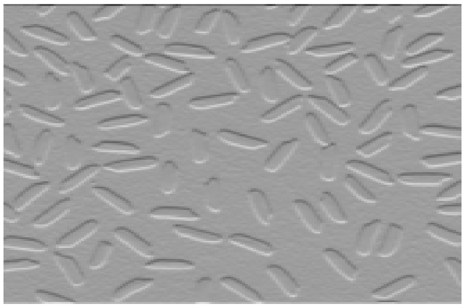


Figure 5. The Plain Image of Rice.

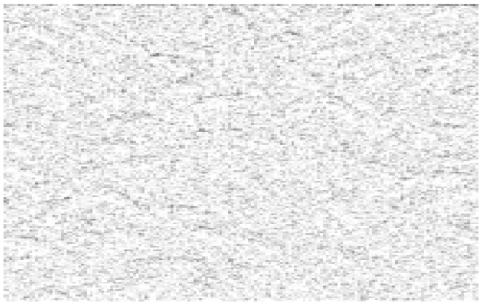


Figure 6. The Cipher Image.

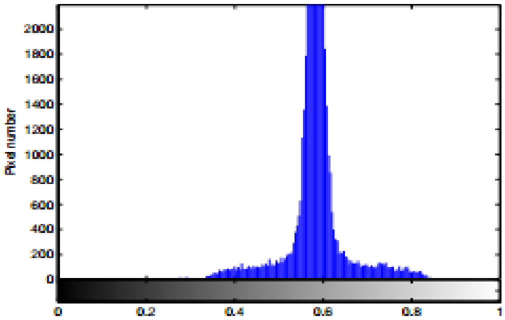


Figure 7. Histogram of Plain Image.

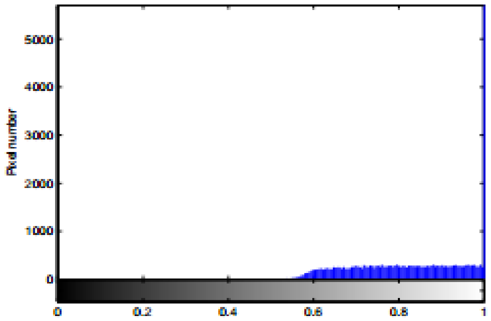


Figure 8. Histogram of Cipher Image.

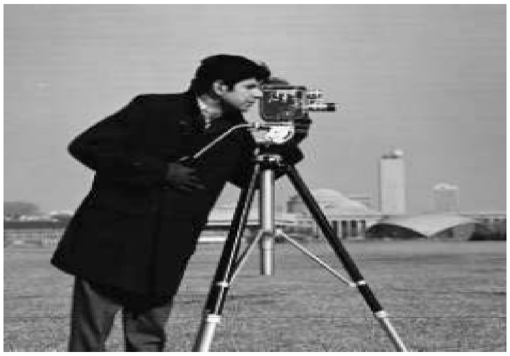


Figure 9. The Plain Image of cameraman.

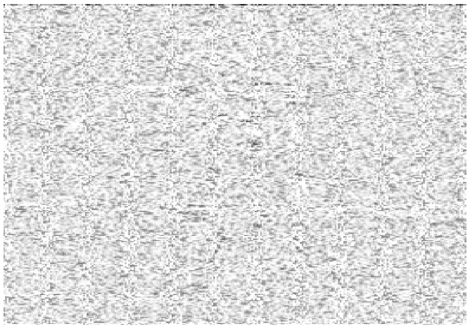


Figure 10. The Cipher Image of cameraman.

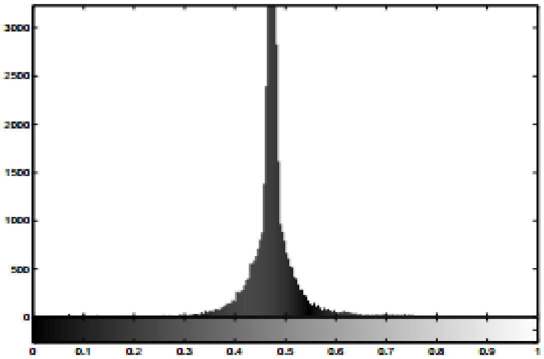


Figure 11. Histogram of Plain Image.

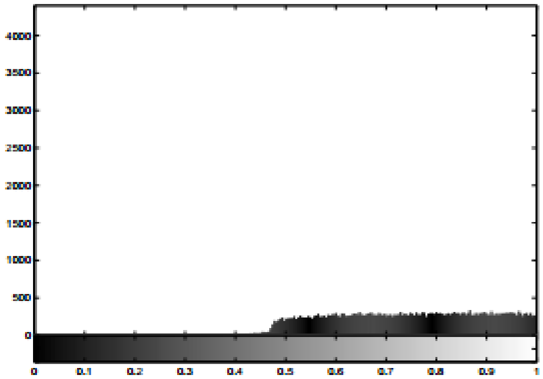


Figure 12. Histogram of Cipher Image.

Table 1. The SNR value of different pictures

Plain Image	256 x 256
Cameraman	0.4361
Rice	0.3621
Lena	0.4798
Tree	0.3123
APC	0.4291
TestPark	0.3982
Elain	0.4853
Truck	0.4663
Tiffany	0.4781
Ruler	0.2216
Couple	0.3672
Aerial	0.3144
Chemical Plant	0.4462
Moon Surface	0.3698

Table 2. Correlation of two adjacent pixels

Correlation	Horizontal	Vertical	Diagonal
Pepper	0.942848	0.945174	0.897210
Encrypted Pepper	–0.000182	0.000357	0.004215
Cameraman	0.933475	0.959223	0.908663
Encrypted Cameraman	–0.000090	–0.007362	0.003039
Rice	0.933471	0.959124	0.908666
Encrypted Rice	0.000012	0.000321	0.003452
Lena	0.904267	0.906432	0.875651
Encrypted Lena	–0.000167	0.000342	0.004875
Jelly beans	0.863571	0.866551	0.824165
Encrypted Jelly beans	–0.000159	0.000332	0.003452
Baboon	0.986453	0.988587	0.93129
Encrypted Baboon	–0.000220	0.000339	0.004876

is quite uniform and different from the original image. Correlation coefficient of two adjacent pixels of both plain image and encrypted image have a fair comparison of all other algorithms.

$$E(x) = \frac{1}{L}\sum_{i=1}^{L} x_i$$

$$D(x) = \frac{1}{L}\sum_{i=1}^{L}[x_i - E(x)]$$

$$Conv(x, y) = \frac{1}{L}\sum_{i=1}^{L}[x_i - E(x)][y_i - E(y)]$$

$$\gamma_{xy} = \frac{Conv(x, y)}{\sqrt{D(x)\sqrt{D(y)}}}$$

This has to be carried out in vertical, horizontal and diagonal directions using the above formulas

4 Conclusion

This paper has proposed a symmetric key cryptographic system using PWLCM, MPWLCM and the genetic algorithm to encrypt plain images. This system, with the use of genetic algorithm for the selection of pixel's position is useful and provided strong cryptographic security. Texts could also be embedded into images and the same has equally employed to RGB images.

5 References

1. Vijayaraghavan R, Sathya S, Raajan, NR. Security for an Image Using Bit-slice Rotation Method-image Encryption. Indian Journal of Science and Technology. 2014 April; 7(Supplementary 4).
2. Tamilselvi R, Ravindran G. Image encryption using Pseudo Random Bit Generator Based on Logistic Maps with Random Transform. Indian Journal of Science and Technology. 2015 June; 8(11); doi: 10.17485/ijst/2015/v8i11/71763.
3. Bano Mahwish, Shah Tariq, Shah Tasneem. Piecewise Linear Chaotic Map Based Image Encryption. International Journal of Machine Learning and Cybernetics. 2015; (In review).
4. Hu Y, Zhu C, and Wang Z. An Improved Piecewise Linear Chaotic Map Based Image Encryption Algorithm. The Scientific World Journal. 2014.
5. Liu H and Wang X. Color image encryption based on one-time keys and robust chaotic maps. Computers and Mathematics with Applications. 2010; 59(10):3320-327.
6. Tong X and Cui M. Image encryption with compound chaotic sequence cipher shifting dynamically. Image and Vision Computing. 2008; 26(6):843-50.
7. Sun F, Liu S, Li Z, and Lu Z. A novel image encryption scheme based on spatial chaos map. Chaos, Solitons and Fractals. 2008; 38(3):631-40.
8. Liu Z, Guo Q, Xu L, Ahmad MA, and Liu S. Double image encryption by using iterative random binary encoding in gyratordomains. Optics Express. 2010; 18(11):12033-2043.
9. Zhang G and Liu Q. A novel image encryption method based on total shuffling scheme. Optics Communications. 2011; 284(12):2775-780.
10. Shannon CE. Communication theory of secrecy systems. Bell Sytem Technical Journal. 1949; 28(4):656-715.
11. Liu H and Wang X. Color image encryption using spatial bit-level per-mutation and high-dimension chaotic system. Optics Communications. 2011; 284(16-17):3895-3903.
12. Wang X and Jin C. Image encryption using game of life permutation and PWLCM chaotic system. Optics Communications. 2012; 285(4):412-17.
13. Indhumathi S, Venkatesan D. Improving coverage Deployment for Dynamic Nodes using Genetic Algorithm in Wireless Sensor Networks. Indian Journal of Science and Technology. 2015; 8(16).