Image Encryption using Pseudo Random Number Generators

Arihant Kr. Banthia
Postgraduate student (MTech)
Deptt. of CSE & IT, MANIT, Bhopal

Namita Tiwari
Asst. Professor
Deptt. of CSE & IT, MANIT, Bhopal

ABSTRACT

Image encryption is conversion of image to a distorted form so that it can be secured from unauthorized users. This paper implements and investigates two methods for image encryption. First technique is encryption of image by linear congruential generator. Random numbers are generated by Linear congruential generator. These numbers are used as index for shuffling of rows, columns and pixels of an image. Second technique uses logistic maps to generate random number sequences. These random numbers are used as index for shuffling of rows, columns and pixels of an image. Finally we have analyzed two methods on basis of image quality parameters.

General Terms

Image Encryption

Keywords

Encryption, Logistic map, Linear congruential generator

1. INTRODUCTION

Now-a-days, digital images are frequently used on internet. Images are used in various fields like military, medical imaging, personal photo security on public networks etc. All existing encryption techniques are mostly for textual data and not suitable for multimedia data such as images, video etc due to three reasons: 1. They are larger size files as compared to text files. 2. They require real time constraints means extracted data should retrieve in particular time interval.3. Its perception i.e. a small distortion in decrypted image is acceptable while in case of textual data it is not possible.

Image Encryption [1] is the term used for conversion of an image from its original form to ciphered form while retrieving original image from its ciphered form is called Image Decryption. Pseudorandom numbers are numbers that are not truly random but appear to be random, produced by some mathematical system or algorithm called Pseudorandom Number Generator (PRNG). Pseudorandom numbers are generated by providing an arbitrary seed value to pseudorandom number generator[2], so later, if same sequence is needed then it can be generated by providing same seed value. So PRNG have a deterministic nature. Another property of PRNG is that number sequence repeats itself. It shows their periodic nature.

Here in this paper, two techniques for image encryption are presented: 1. Image encryption by linear congruential generator. 2. Image Encryption based on chaotic logistic map. Both techniques use basic cryptography operations i.e. permutation and substitution. Permutation is done by shuffling of rows, columns and pixels. Substitution is done by masking operation between two adjacent rows and columns.

This paper will evaluate results on following image quality parameters:

(a) Entropy: Entropy [1] [2] is the measure of unpredictability of information content, i.e. more the entropy is then the data will be more disordered. In an encrypted image, the entropy should be as high as possible so that prediction of information becomes difficult.

(b) Cross Correlation: Cross correlation [1] [3] [4] is defined as relationship of corresponding pixels between original image and its ciphered image. If cross correlation value is higher, it means more number of pixels at corresponding positions have similar value. In an encrypted image, the cross correlation value should be as low as possible.

(c)Mean Square Error (MSE): Error is difference of pixel value of original image to pixel value of encrypted image. Mean square error[5] is calculated by taking average of square value of error.

(d) Peak Signal to noise ratio (PSNR): PSNR[2] is defined as ratio of amount of significant signal information to noise. This parameter shows quality measure of an encryption technique. Lower the value of PSNR, more the encryption is stronger because it shows resultant cipher image is noise like and it contains very less amount of significant information.

The rest of this paper is as follows. In section two, both techniques are discussed briefly and their results on different input parameters are shown .In section three analysis of both techniques are done based on quality measurement parameters. Last section contains conclusion.

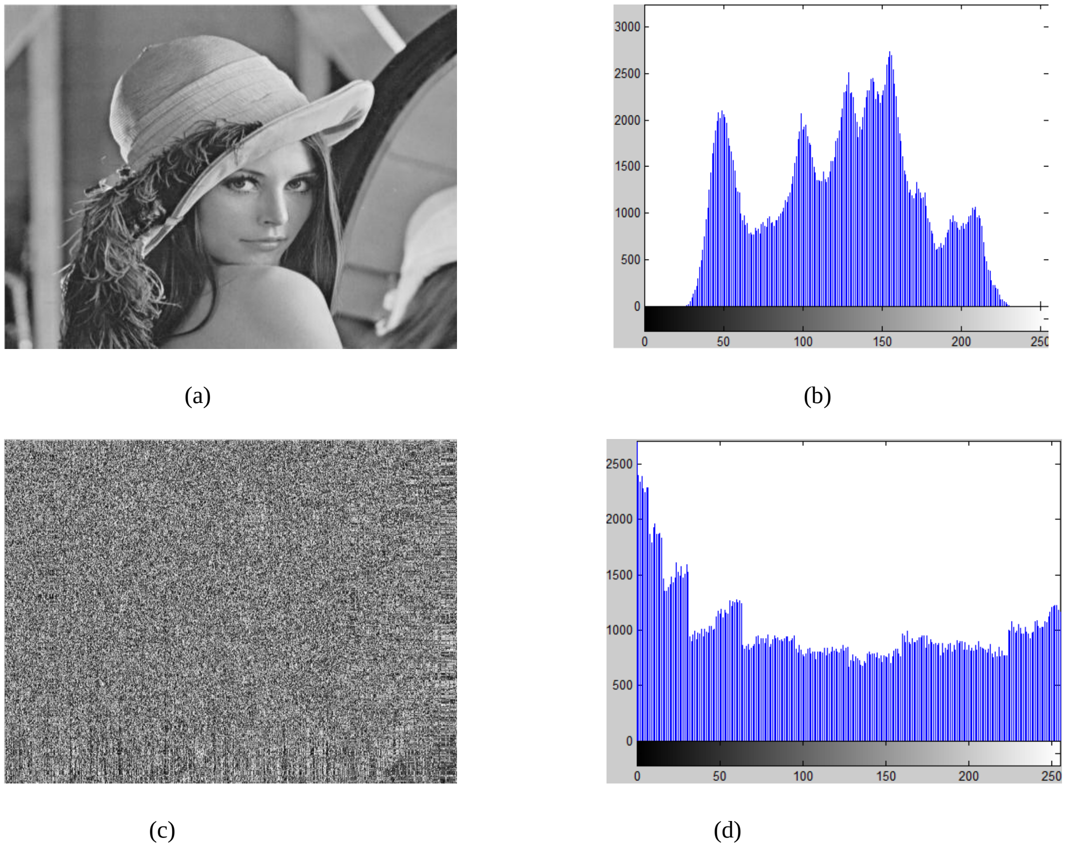


Figure 1 (a) : Original Lena image (b) Histogram of Lena image(c)encrypted image of lena by ImageEncryptLCG() (d) Histogram of encrypted lena image

2. IMAGE ENCRYPTION TECHNIQUES

2.1 Image Encryption using Linear Congruential Generator :

It is most commonly used method for pseudo number generation [2], defined by following equation:

$X_{n+1} = (aX_n+c) \bmod m$ ----- (1)

Where

a- multiplier
m- Modulus
c- Constant to be added

An arbitrary starting seed value(X_0) is needed in equation(1) with above mentioned parameters for generation of random numbers which have range up to the value of modulus (m).In this scheme, two random numbers sequences are generated based on equation(1), by choosing appropriate parameters and seed value. Then by using values of these random numbers, image permutation occurs by shuffling of rows, columns and pixels of image. One sequence is used for row shuffling and another is used for column shuffling. A masking operation [6] is used after row and column shuffling by simple XOR operations between adjacent rows and columns. By values of both sequences, pixel shuffling is done. The whole operation may be summarized by this equation:

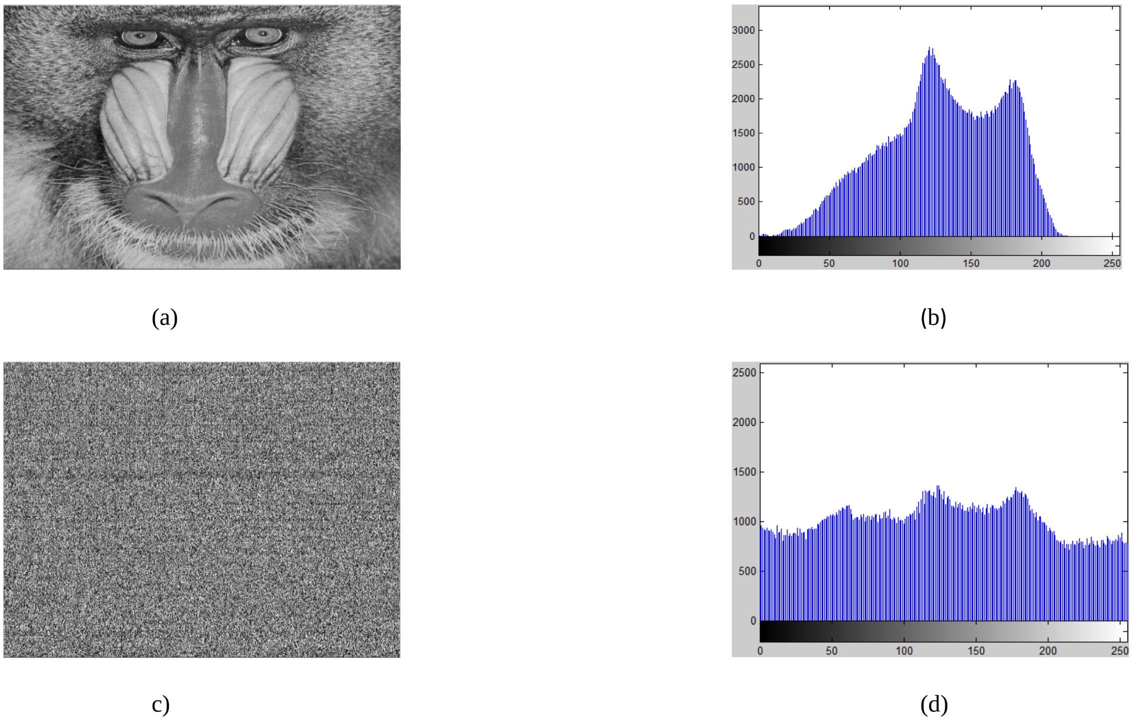
Cimg = Epixel (Ecolumn (Erow (plainim))) ----- (2)

Where

Erow – Encryption by row shuffling and masking
Ecolumn – Encryption by column shuffling and masking
Epixel – Encryption by pixel shuffling.

2.1.1. Pseudo Code : ImageEncryptLCG()

```
Input image, secret key which contains input parameters for linear congruential generator as well as seed values for two sequences.
//Permutation and masking of Rows
i = 1, j = 1
while i < number of rows
generate $x_i$ sequence for i by equation (1).
if $x_i$ > number of rows
increase i , do nothing and continue.
Else
Increase j.
$xoriginal_j = x_i$
Swap $j^{th}$ row by $xoriginal_j^{th}$ row.
Mask $j^{th}$ row with $(j+1)^{th}$ row by applying XOR operations.
End if.
End while.
// Permutation and masking of Column
i=1,k = 1
while k < number of columns
generate $y_i$ sequence for every i.
if $y_i$ > number of columns
increase i, do nothing and continue.
Else
Increase k.
```



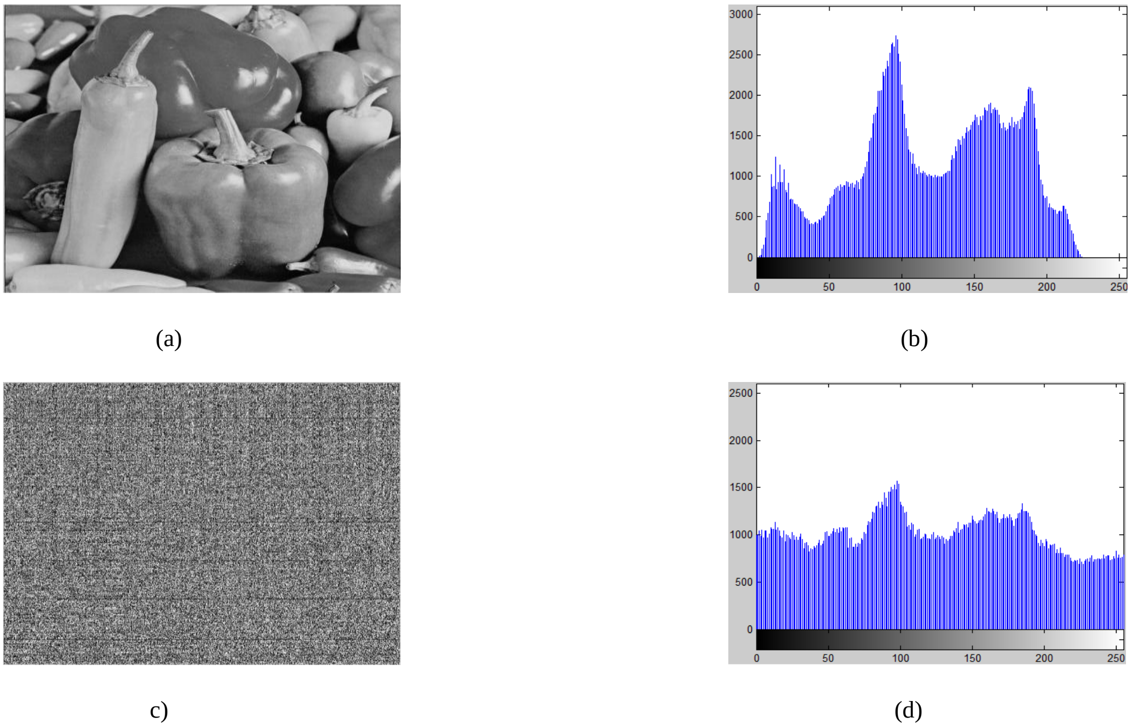
(a)

(b)

c)

(d)

Figure 2 (a): Original Baboon image (b) Histogram of Baboon image(c) encrypted image of Baboon by ImageEncryptLCG () (d) Histogram of encrypted Baboon image



(a)

(b)

c)

(d)

Figure 3 (a): Original Peppers image (b) Histogram of Peppers image(c) encrypted image of Peppers by ImageEncryptLCG () (d) Histogram of encrypted Peppers image

Table 4 Result for image - lena.tif by ImageEncryptLCG ()

a	M	Seed(x0,y0)	Entropy	Cross-corr	MSE	PSNR
92717	262139	(44,44)	7.9617	0.0126	6.86E+03	9.768
118068	262139	(97,44)	7.9618	0.0019	6.91E+03	9.7386
166972	262139	(44,44)	7.9533	0.0039	6.80E+03	9.8047
283741	524287	97,44	7.9641	0.005	6.93E+03	9.7234
37698	524287	97,44	7.9665	0.0069	7.03E+03	9.6601
178	251	44,44	7.9716	-0.0101	7.43E+03	9.4206

Table 5 Result for image - baboon.tif by ImageEncryptLCG ()

a	m	Seed(x0,y0)	Entropy	Cross-corr	MSE	PSNR
92717	262139	(44,44)	7.9818	0.0056	6.62E+03	9.9203
118068	262139	(97,44)	7.986	0.0077	6.68E+03	9.8799
166972	262139	(44,44)	7.9791	0.013	6.50E+03	10.001
283741	524287	(97,44)	7.9857	0.009	6.68E+03	9.8859
37698	524287	(97,44)	7.9882	0.0105	6.72E+03	9.8551
178	251	(44,44)	7.9884	0.0127	7.05E+03	9.6516

Table 6 Result for image - peppers.tif by ImageEncryptLCG ()

a	m	Seed(x0,y0)	Entropy	Cross-corr	MSE	PSNR
92717	262139	(44,44)	7.9762	-1.60E-05	7.83E+03	9.1919
118068	262139	(97,44)	7.9747	0.0024	7.83E+03	9.1937
166972	262139	(44,44)	7.9689	0.0035	7.69E+03	9.2699
283741	524287	(97,44)	7.9759	2.83E-04	7.81E+03	9.2065
37698	524287	(97,44)	7.9796	0.003	7.88E+03	9.1666
178	251	(44,44)	7.983	-0.0228	8.17E+03	9.0072

```
22. Swap $k^{th}$ column by $yoriginal_k{}^{th}$ column.
23. Mask $k^{th}$ column with $(k+1)^{th}$ column by applying XOR
    operations.
24. End if
25. End while.
// Permutation of Pixels
26. j = 1, k = 1
27. while j < number of rows and k < number of columns.
28. swap $Img(j,k)^{th}$ pixel with $Img(xoriginal_j,yoriginal_k)^{th}$
    pixel.
29. End while.
```

Results: Results are calculated on standard images by taking different values of a, m, seed x_0 and seed y_0, which are supplied as part of key. Results are shown in table (1), (2), (3) for different images. Three different images with their encrypted form and their histograms are shown in figure 1, 2, 3.

2.2 Image Encryption using Chaotic logistic map :

Logistic map is a mathematical iterative system used for generating random numbers, defined by following iterative equation:

$$X_{n+1} = r*X_n*(1-X_n) \text{ --- } (3)$$

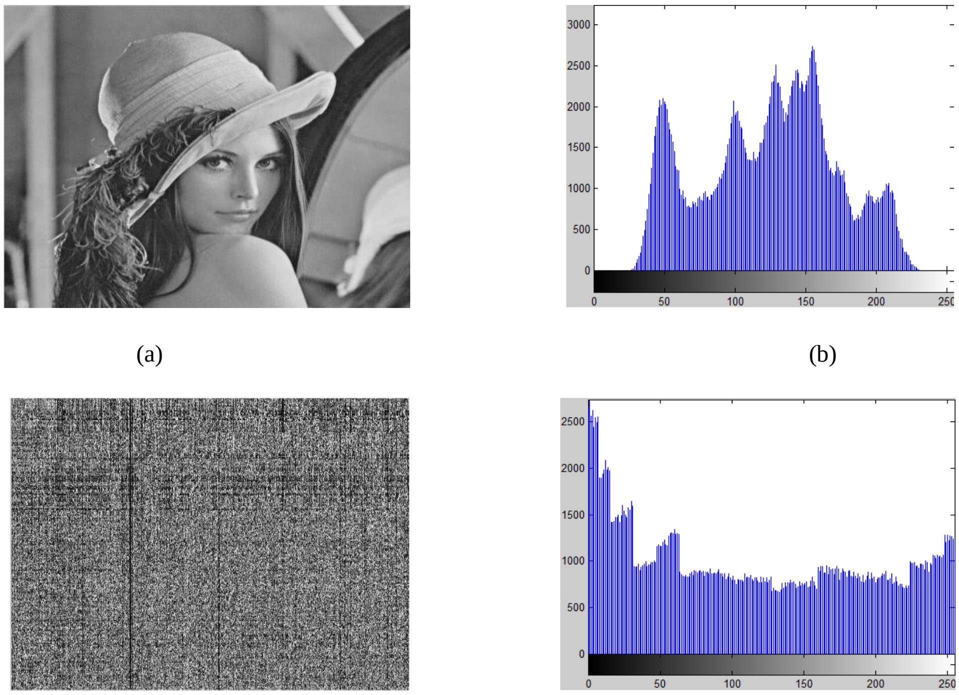
Where r is growth rate parameter. By choosing appropriate seed value (X_0) and growth rate (r), equation (3) can be used to generate random number sequence [2] [6] [7] which have long period value. In this scheme, two random numbers sequences are generated based on chaotic logistic map. One sequence is used for row shuffling, another for column shuffling. Pixel shuffling is done by taking both sequences together, same as scheme (A). A masking operation[8] is used after row and column shuffling by simple XOR operations between adjacent rows and columns. The whole operation may be summarized, same as scheme (A) by this equation:

Cimg = Epixel (Ecolumn (Erow (plainim))) ----- (4)

Where
Erow – Encryption by row shuffling and masking
Ecolumn – Encryption by column shuffling and masking
Epixel – Encryption by pixel shuffling.

2.2.1. Pseudo Code: ImageEncryptionChaos()

```
1.  Input image, secret key which contains parameters for
    logistic map as well as seed values for two sequences.
//Permutation and masking of Rows
2.  i = 1, j = 1
3.  while j < number of rows
4.  generate $x_i$ sequence for i by equation(2)
5.  Convert real value $x_i$ in integer.
6.  if $x_i$ > number of rows
7.  increase i , do nothing and continue
8.  Else
9.  Increase j.
10. $xoriginal_j = x_i$
11. Swap $j^{th}$ row by $xoriginal_j{}^{th}$ row.
12. Mask $j^{th}$ row with $(j+1)^{th}$ row by applying XOR
    operations.
13. End if.
14. End while
// Permutation and masking of Column
15. i=1,k = 1
```



(a) (b)

(c) (d)

Figure 4 (a): Original Lena image (b) Histogram of Lena image

(c) encrypted image of lena by ImageEncryptionChaos () (d) Histogram of encrypted lena image

```
16. while k < number of columns
17. generate yi sequence for every i by equation(2)
18. convert real value yi in integer.
19. if yi > number of columns
20. increase i, do nothing and continue.
21. Else
22. Increase k.
23. yoriginalk = yi.
24. Swap kth column by yoriginalkth column.
25. Mask kth column with (k+1)th column by applying XOR
    operations.
26. End if.
27. End while
// Permutation of Pixels
28. j = 1, k = 1
29. while j < number of rows and k < number of columns
30. swap Img (j, k)th pixel with Img(xoriginalj,yoriginalk)th
    pixel.
31. End while
```

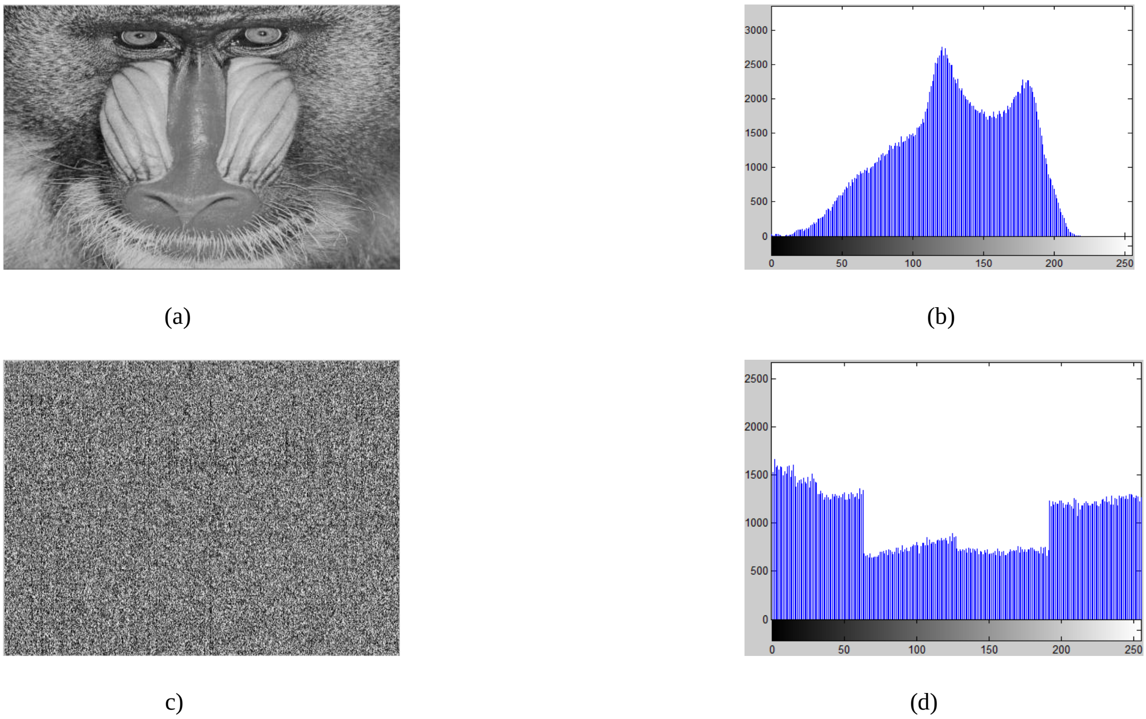
Results: Results are calculated on standard images by taking different values of r, seed x_0 and seed y_0, which are supplied as part of key. Results are shown in table (4),(5),(6) for different images. Three different images with their encrypted form and their histograms are shown in figure 4, 5, 6.

3. RESULT ANALYSIS

From the above tables we can analyze results of both techniques. Both techniques are giving good results on selected values of input parameters. A table is showing range of output values which are produced on above three images. From the result we can conclude that best output parameters values for different images depend on different input parameters. ImageEncryptLCG() gives better result at a = 37698, m = 524287, (x_0,y_0) = (97,44)for achieving high Entropy; a = 92717, m = 262139, (x_0,y_0) = (44,44) for low correlation; a = 178, m = 251, (x_0,y_0) = (44,44) high MSE and low value of PSNR for lena image. ImageEncryptionChaos() produces better results at r = 3.8, seed $x_0 = 0.3427$,$y_0 = 0.4256$ for high entropy, r = 3.78, seed $x_0 = 0.7234$, $y_0 = 0.8694$ for low cross correlation, r =3.67, seed x_0 =0.8445, seed y_0 =0.9345 for high MSE and low PSNR for same image. The comparison of two existing techniques by showing the range of output parameters produced by each of them has been summarized in table 7 for different images.

4. CONCLUSION

In this paper, analysis of two highly secure techniques of image encryption using pseudorandom number generators is done. Both techniques encrypt images by permutation and substitution operations and calculate some parameters value as results. The performance of both algorithms are good but it is to be mentioned that the results obtained for each algorithm are confined to specific input parameters used across the experiments. Based on the choice of the input parameters, which are part of secret key, both algorithm may exhibit similar or different results, without rendering any effect on the security of the image under encryption. Further for more better results both techniques can be merged and some more masking operations can be applied by choosing particular seed rows as well as seed columns as part of key. This will increase the length of key and may improve results further.



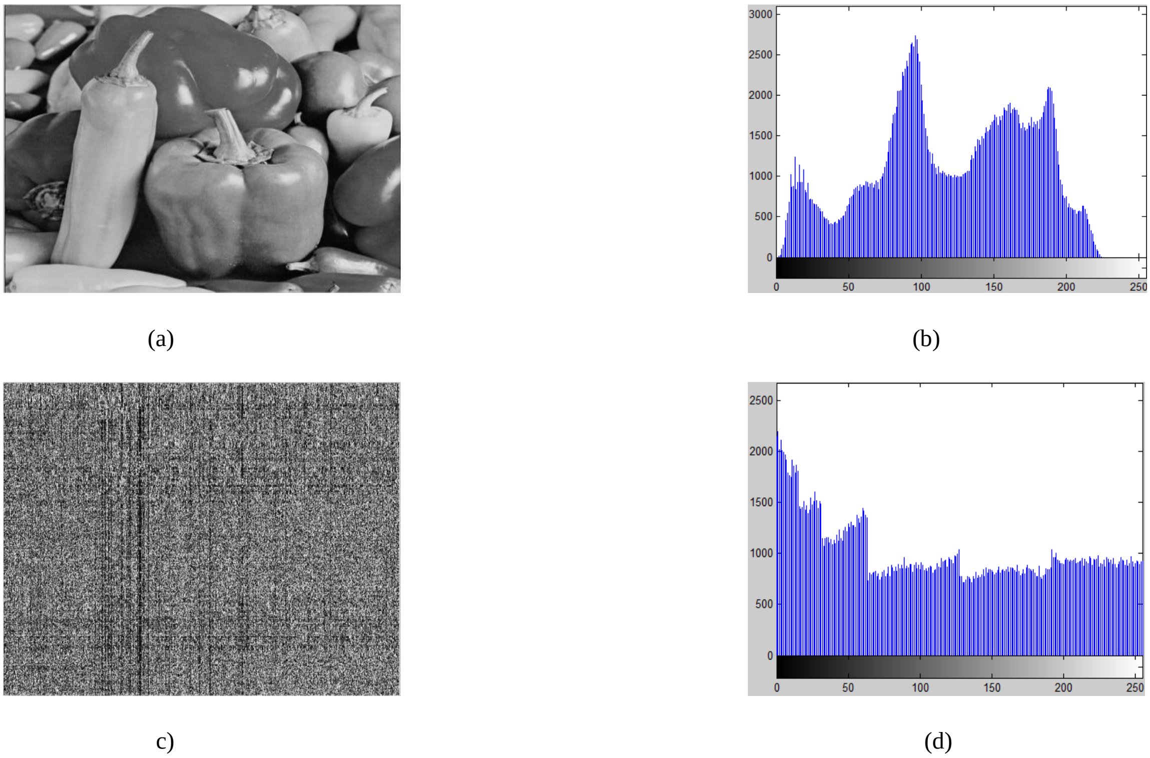
(a)

(b)

c)

(d)

Figure 5 (a) : Original Baboon image (b) Histogram of Baboon image(c)encrypted image of Baboon by ImageEncryptionChaos() (d) Histogram of encrypted Baboon image



(a)

(b)

c)

(d)

Figure 6 (a) : Original Peppers image (b) Histogram of Peppers image(c)encrypted image of Peppers by ImageEncryptionChaos() (d) Histogram of encrypted Peppers image

Table 10 Results for image : lena.tif by ImageEncryptionChaos()

r	Seed x_0	Seed y_0	Entropy	Cross-corr	MSE	PSNR
3.78	0.7234	0.8694	7.9151	-0.0063	8.99E+03	8.5927
3.78	0.7234	0.7286	7.9147	-0.0011	8.95E+03	8.6109
3.78	0.8445	0.9345	7.9219	9.72E-05	8.90E+03	8.6391
3.8	0.3427	0.4256	7.9325	-0.005	8.87E+03	8.6502
3.67	0.8445	0.9345	7.8837	0.0036	9.16E+03	8.5112

Table 11 Result for image - Baboon.tif by ImageEncryptionChaos()

r	Seed x_0	Seed y_0	Entropy	Cross-corr	MSE	PSNR
3.78	0.7234	0.8694	7.934	-0.0053	8.59E+03	8.7931
3.78	0.7234	0.7286	7.9362	-0.0034	8.56E+03	8.8077
3.78	0.8445	0.9345	7.9372	-0.0059	8.55E+03	8.8137
3.8	0.3427	0.4256	7.9357	0.0037	8.52E+03	8.8287
3.67	0.8445	0.9345	7.9061	-2.90E-04	8.84E+03	8.6656

Table 12 Results for Peppers.tif by ImageEncryptionChaos()

r	Seed x_0	Seed y_0	Entropy	Cross-corr	MSE	PSNR
3.78	0.7234	0.8694	7.9447	-0.0043	9.19E+03	8.498
3.78	0.7234	0.7286	7.9439	-5.07E-04	9.13E+03	8.5246
3.78	0.8445	0.9345	7.945	-7.73E-04	9.15E+03	8.5189
3.8	0.3427	0.4256	7.9464	0.0054	9.09E+03	8.5445
3.67	0.8445	0.9345	7.905	-0.0053	9.48E+03	8.3631

Table 7 Comparison of two techniques

	Technique	Entropy	Cross-Corr	MSE	PSNR
Image – Lena.tif	ImageEncryptionLCG()	7.95 to 7.97	-0.01 to 0.01	6.80E+03 to 7.43E+03	9.42 to 9.80
	ImageEncryptionChaos()	7.88 to 7.91	-0.006 to 0.003	8.87E+03 to 9.16E+03	8.51 to 8.65
Image – Baboon.tif	ImageEncryptionLCG()	7.98 to 7.99	0.005 to 0.01	6.50E+03 to 7.05E+03	9.65 to 10
	ImageEncryptionChaos()	7.91 to 7.94	-2.90E-04 to 0.0037	8.52E+03 to 8.84E+03	8.67 to 8.83
Image – Peppers.tif	ImageEncryptionLCG()	7.96 to 7.98	-1.60E-05 to 0.0035	7.69E+03 to 8.17E+03	9 to 9.27
	ImageEncryptionChaos()	7.9 to 7.95	-7.73E-04 to 0.0054	9.09E+03 to 9.48E+03	8.36 to 8.54

5. REFERENCES

[1] Syscom Laboratory, Ecole Nationale d’Ingénieurs de Tunis,Tunisia,”Security analysis of image cryptosystems only or partially based on a chaotic permutation”, The Journal of Systems and Software 85(2012) 2133– 2144.

[2] G.A.Sathishkumar, Srinivas Ramachandran, Dr.K.Bhoopathy Bagan “Image Encryption Using Random Pixel Permutation by Chaotic Mapping” 2012 IEEE Symposium on Computers & Informatics, 2012.

[3] Bin Wang, Xiaopeng Wei, Qiang Zhang,” Cryptanalysis of an image cryptosystem based on logistic map”, Optik xxx (2012) xxx–xxx

[4] Mohammad Ali Bani Younes and Arnan Jantan ,” Image Encryption Using Block-Based transformation Algorithm”, IAENG International Journal of Computer Science, 35:1, IJCS_35_1_03.

[5] Vibha Tiwari, P.P. Bansod and Abhay Kumar, “Performance Evaluation of Various Compression Techniques on Medical Images”, International journal of advanced electronics & communication systems, Issue 2 Volume 1 May 2012.

[6] Jiankun Hu, Fengling Han, “A pixel-based scrambling scheme for digital medical images protection” Journal of Network and Computer Applications 32 (2009) 788–794.

[7] Hossam El-din H. Ahmed, Hamdy M. Kalash, and Osama S. Farag Allah ,” An Efficient Chaos-Based Feedback Stream Cipher (ECBFSC) for Image Encryption and Decryption”, Informatica 31 (2007) 121–129.

[8] Koredianto Usman, Hiroshi Juzojil IsaoNakajimal, Soegijardjo Soegidjoko, Mohamad Ramdhani Toshihiro Hori, SeijiIgi "Medical image encryption based on pixel arrangement and random permutation for transmission security" 1-4244-0942-X/07/ 2007IEEE.

[9] Akhavan, A. Samsudin, A. Akhshani, "A symmetric image encryption scheme based on combination of nonlinear chaotic maps" Journal of the Franklin Institute 348 (2011) 1797 – 1813.

[10] Chang C, Hwang M, Chen T, "A new encryption algorithm for image cryptosystems."JSyst Software;58:83–91, science, vol. 809, Springer, Berlin; 1993. p. 71–82, 2001.