Secure Sequences for Pixel Encryption by Coalescing PRNGs and Chaotic maps

M. Y. Mohamed Parvees[*,1]
* Division of Computer & Information Science,
Annamalai University
Annamalainagar – 608002, India
yparvees@gmail.com
(Corresponding author)
[1]Research & Development Centre, Bharathiar University
Coimbatore -641046, India

J. Abdul Samath
Department of Computer Science
Chikkanna Government Arts College
Tiruppur-641602, India
abdul_samath@yahoo.com

B. Parameswaran Bose
Independent Researcher
#35, Indiragandhi Street, Udayanagar
Bengalore, India
parameswaran.gri@gmail.com

***Abstract*— The effective use of random sequences is a essential requirement in day-today cryptographic operations. Particularly, the cryptographically secure random sequences play a vital part in encryption processes. The deterministic randomness process is helpful to provide cryptographic solutions for different types of security primitives, mostly, in order to provide security through encryption. The chaotic equations are having the phenomenon of deterministic randomness. The chaotic maps are generally involving in cryptographic solutions due to their sensitive chaotic outputs towards their tiny variations in its inputs. Though the chaotic maps are effectively involved to enhance security higher, the improved pseudo random number generators (PRNGs) could be used along with the chaotic map during the chaotic sequences generations. These cryptographically secure random sequences can be applied to cryptographic primitives during the confusion-diffusion processes. These sequences are validated through the randomness test suites. The results proves that the sequences generated by coalescing PRNGs and Chaotic equations are cryptographically secure and could be involved in pixel encryption processes.**

Keywords*— Linear Congruential Generator, XOR Shift Generator, Chaotic map, Pseudo Random Number Generator.

I. INTRODUCTION

The demands for random number generation and use of random numbers in day-today life are increasing rapidly. The chaotic maps are highly useful in large file encryptions including image encryption [1-3], audio [4] and video encryptions [5] to provide confidentiality to data. The chaotic maps could be applied for creating basic encryption processes such as confusion, diffusion, swapping, etc [6-7]. The chaotic equations produce chaotic output in a deterministic way. The special property of deterministic randomness helps to use chaotic maps in encryption processes. Yet several researchers involve chaotic maps to generate secure sequences and the sequences are effectively engaged in encryption processes [1-3,6,7]. Similarly, pseudorandom number generators are also used in many cryptographic solutions [8]. Parvees et al employs chaotic maps along with PRNGs to produce random sequences to confuse and diffuse image pixels thereby secure the colour DICOM images [8]. Though, the images are secure, it is essential to check whether the random sequences engaged for encryption processes are cryptographically secure or not. Usually, adversaries try to find out confusion-diffusion sequence, thereby they guess the secret information. Therefore, it is essential to check random sequences for their sternness towards the adversary attacks. The randomness of the sequences directly generated from improved logistic 2 dimensional coupled chaotic map (IL2DCCM), Enhanced Chaotic Economic map are cryptographically secure which can be employed in confusion-diffusion sequence generation algorithms[8-9]. Yet, it is important to check the exact confusion-diffusion sequences for their sternness towards cryptographic attacks. Hence, this study proposes to validate the confusion and diffusion sequences generated by coalescing IL2DCCM with improved XOR shift generator (IXSG) and improved linear congruential generator (ILCG).

II. PRELIMINARIES

A. *Improved Logistic 2D Coupled map*

The improved logistic 2D coupled chaotic map is obtained from basic L2DCCM which is shown in equations (1-2) [8]. It is two dimensional map and able to produce two different chaotic sequences.

$$x_{n+1} = miu_1 \times x_n \times (1 - x_n) + (\gamma_1 \times y_n{}^2) \tag{1}$$

$$y_{n+1} = miu_2 \times y_n \times (1 - y_n) + \left[\gamma_2 \times \left(x_n{}^2 + (x_n \times y_n)\right)\right] \tag{2}$$

where, $x_i \in [0,1)$ is an independent variable; $2.75 < miu_1 \leq 3.40$, $2.75 < miu_2 \leq 3.45$, $0.15 < \gamma_1 \leq 0.21$, $0.13 < \gamma_2 \leq 0.15$ are the control parameters. the equations behaves chaotically and produce two different chaotic sequences. The diagram of bifurcate and Lyapunov are shown Fig. 1-4. The bifurcate range gives the idea of chaotic keys. In order to increase the number of chaotic keys, the bifurcation range could be enlarged thereby attaining higher security towards brute-force attack. So the basic maps are improved and given in equations (3) and (4).

$$x_{n+1} = miu_1 \times \sin(x_n) \times (1 - \sin(x_n)) + \left(\gamma_1 \times (\sin(y_n))^2\right) \tag{3}$$

$$y_{n+1} = miu_2 \times \sin(y_n) \times (1 - \sin(y_n)) + \left[\gamma_2 \times \left((\sin(x_n))^2 + (\sin(x_n) \times \sin(y_n))\right)\right] \tag{4}$$

where, $5.0 < miu_1 \leq 12.0$, $0 < miu_2 \leq 3.45$, $0.15 < \gamma_1 \leq 0.21$, $0.13 < \gamma_2 \leq 0.15$ and give two sequences for $x \in (0, 3.1]$ and $y \in (0, 1]$.

The bifurcation range of improved L2DCCM is higher than the L2DCCM. This enlighten that the improved chaotic map increases the key space. The equation (4) and (5) produces the chaotic sequence. These chaotic sequences can be fused with other PRNGs sequences to produce cryptographically secure sequences required for encryption processes such as confusion, diffusion, swapping, etc.

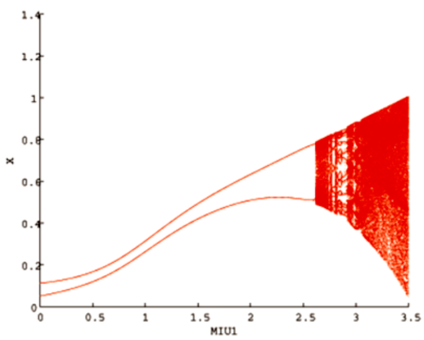

Fig. 1 Bifurcation of LC2DM (x_i)

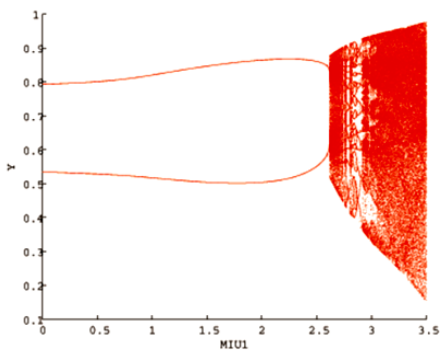

Fig. 2 Bifurcation of LC2DM (y_i)

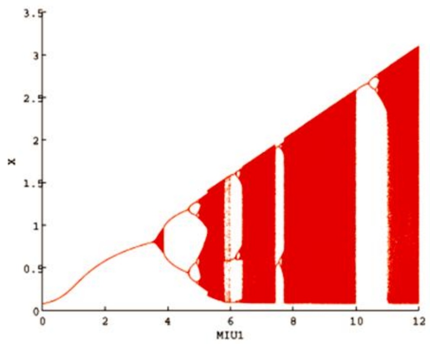

Fig. 3 Bifurcation of ILC2DM (x_i)

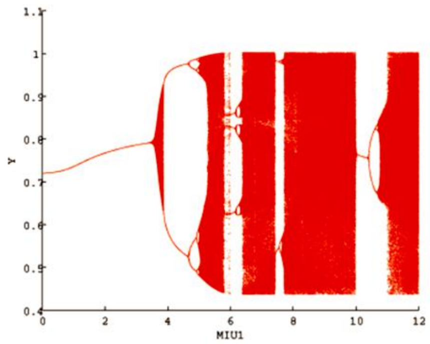

Fig. 4 Bifurcation of ILC2DM (y_i)

B. Improved XOR Shift Generator

The XOR shift generator (XSG) is a speedy pseudo random number generator given in equation (5) [8].

$$x = x_{n-1} \wedge [(x_{n-1}) H_1 a]$$

$$x = x \wedge [(x) H_2 b]$$

$$x_n = \{x \wedge [(x) H_3 c]\} mod\ 2^{32} \tag{5}$$

if $a < 0$, then H_1 is $'>>'$. If $a > 0$, then H_1 is $'<<'$. Similar conditions are employed for H_2 and H_3 where, $-64 < a,b,c < 64$. The three different improved XSG are obtained from basic XSG and given in equations (6-8) [8].

$$x = x_{n-1} \wedge [(x_{n-1}) H_1 a]$$

$$x = x \wedge [(x) H_2 b]$$

$$x_n = \{(i + x) \wedge [(x) H_3 c]\} mod(m) \tag{6}$$

$$x = (i \times x_{n-1}) \wedge [i + (x_{n-1}) H_1 a]$$

Pseudocode 1: Random sequences generation from IXSG

```
1: BEGIN
2: Initialise input parameters for ILCG
3: SET number_of_parameters ← 6
4: SET seq_no ← 127000000
5: SET a← 13
6: SET b← -17
7: SET pre← 2147480000
8: SET c← 5
9: SET m← 16777215
10: Generate random sequences using the equation (6)
11: Method: genRandomSequences(seq_no, a, b, pre, c,m)
12: INPUT: Parameters seq_no, a, pre, c,m
13: seq[0] ← pre
14: for i←1 to seq_no do
15: x ← seq[i - 1] ^ (a > 0 ? (seq[i - 1] << Math.abs(a)) : (seq[i - 1] >> Math.abs(a)));
    x ←x ^(b > 0 ? (x << Math.abs(b)) : (x >> Math.abs(b)));
    seq[i] ← (int) (((i + x) ^ (c > 0 ? (x << Math.abs(c)) : (x >> Math.abs(c)))) % m);
16. Return seq
17: end for
18: END
```

$$x = x \wedge \left[i + (x) H_2 b \right]$$

$$x_n = \left\{ (i \times x) \wedge \left[i + (x) H_3 c \right] \right\} mod\,(m) \quad (7)$$

$$x = (i \times x_{n-1}) \wedge \left[i + (x_{n-1}) H_1 a \right]$$

$$x = x \wedge \left[i + (x) H_2 b \right]$$

$$x_n = \left\{ x \wedge \left[i + (x) H_3 c \right] \right\} mod\,(m) \quad (8)$$

where, x_n is random integer, m is modulus number, a,b,c are bit shift numbers and $i \in (0,n)$.

C. Improved LCG

The Linear Congruential generator (LCG) is also a speedy pseudo random number generator given in equation (9) [8].

$$x_{n+1} = \left[(a \times x_n) + c \right] mod\ m \quad (9)$$

The three different improved LCG are also obtained from the basic LCG and given in equations (10-12) [8].

$$x_{n+1} = \left\{ i \times \left[(a \times x_n) + (i + c) \right] \right\} mod\ m \quad (10)$$

$$x_{n+1} = \left\{ i \times \left[(a \times x_n) + (i \times c) \right] \right\} mod\ m \quad (11)$$

$$x_{n+1} = \left\{ i \times \left[(a \times x_n) + (i \wedge c) \right] \right\} mod\ m \quad (12)$$

where x_n is random integer, m is modulus number, a is the multiplying integer with the interval $(0,m)$. c decides order of random numbers.

III. METHODOLOGY

The idea is to create confusion and diffusion sequences for encrypting pixels. The encryption usually involves pixel and byte confusion-diffusion processes. This study focus only on the importance of sequences generated for pixel confusion and diffusion processes. The primary objective is to study the randomness of the confusion-diffusion sequences thereby proving that they are cryptographically secure and the sequences can be effectively involved in encryption processes. Since the colour images have the 24-bit values with RGB separation, the random numbers are generated from the IXSG, ILCG and coalesced with the sequences generated from IL2DCCM.

Initially, the three different 24-bit random numbers N1, N2, N3 are produced using IXSG equations (6-8) as given in Pseudocode 1. The random matrix is a complex one and obtained using the following steps.

```
R1=((N1>>16)&0xFF)^((N2>>08)&0xFF)^((N2)&0xFF)^((N3>>08)&0xFF)^((N3)&0xFF)
G1=((N1>>08)&0xFF)^((N2>>16)&0xFF)^((N2)&0xFF)^((N3>>16)&0xFF)^((N3)&0xFF)
B1=((N1)&0xFF)^((N2>>16)&0xFF)^((N2>>08)&0xFF)^((N3>>16)&0xFF)^((N3>>08)&0xFF)
R2=((N2>>16)&0xFF)^((N1>>08)&0xFF)^((N1)&0xFF)^((N3>>08)&0xFF)^((N3)&0xFF)
G2=((N2>>08)&0xFF)^((N1>>16)&0xFF)^((N1)&0xFF)^((N3>>16)&0xFF)^((N3)&0xFF)
B2=((N2)&0xFF)^((N1>>16)&0xFF)^((N1>>08)&0xFF)^((N3>>16)&0xFF)^((N3>>08)&0xFF)
R3=((N3>>16)&0xFF)^((N1>>08)&0xFF)^((N1)&0xFF)^((N2>>08)&0xFF)^((N2)&0xFF)
G3=((N3>>08)&0xFF)^((N1>>16)&0xFF)^((N1)&0xFF)^((N2>>16)&0xFF)^((N2)&0xFF)
B3=((N3)&0xFF)^((N1>>16)&0xFF)^((N1>>08)&0xFF)^((N2>>16)&0xFF)^((N2>>08)&0xFF)
Pix1=(R1<<16)+(G1<<8)+B1
```

Pix2=(R2<<16)+(G2<<8)+B2

Pix3=(R3<<16)+(G3<<8)+B3

$$XSG_{pixels} = Pix_1 \wedge Pix_2 \wedge Pix_3$$

The XSG_{pixels} can be used for confusion process along with the sequences of IL2DCCM. Similarly, the LCG_{pixels} are also obtained by iterating the equations (10-12) as given in Pseudocode 2 and following the above similar steps. LCG_{pixels} could be used for diffusing the pixels along with IL2DCCM. Further, two different chaotic sequences, namely, $x_{sequences}$, $y_{sequences}$ are generated from IL2DCCM using the equations (3 & 4) and given in Pseudocode 3. The actual coalescing the sequences of PRNGs' and IL2DCCM occur in the following equations.

$$confusionseq_{pixels} = \text{int}\left[\left(\left(abs(x_{sequences}) \times i \times 10^{8}\right) + XSG_{pixels}\right) mod\ 12700000\right]$$

$$diffusionseq_{pixels} = \text{int}\left[\left(abs(y_{sequences}) \times 10^{16} + LCG_{pixels}\right) mod\ (2^{24} - 1)\right]$$

From the equations, the confusion sequences are generated by coalescing the $x_{sequences}$ from IL2DCCM and XSG_{pixels} from IXSG. Similarly, the diffusion sequences are generated by coalescing the $y_{sequences}$ from IL2DCCM and LCG_{pixels} from ILCG. It is essential to validate the $confusionseq_{pixels}$ and $diffusionseq_{pixels}$ for their randomness in order to find whether they are cryptographically secure or not.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The confusion and diffusion sequences are validated for their randomness through the NIST STS-2.1.2 [10], Diehard [11] and Entropy [12] test suites. The $confusionseq_{pixels}$ and $diffusionseq_{pixels}$ for the size of 127000000 were generated and validated for randomness. The corresponding bits and byte files are created from the $confusionseq_{pixels}$ and $diffusionseq_{pixels}$. Table 1 and 2 yields the successful test results which prove that the $confusionseq_{pixels}$ and $diffusionseq_{pixels}$ are cryptographically secure that are generated by coalescing the PRNGs with chaotic map. The results were optimum while comparing with similar studies [13-16]. Similarly, the $confusionseq_{pixels}$ and $diffusionseq_{pixels}$ are validated for randomness using Diehard test suite and the results are listed in Table 3 and 4. The sequences pass all the diehard tests also. Further, $confusionseq_{pixels}$ and $diffusionseq_{pixels}$ are also validated for randomness towards Entropy test suites and Table 4 and 5 shows the results of Entropy tests. The p-values of all tests prove that the sequences are cryptographically secure enough to use in any cryptographic operations such as confusion-diffusion in encryption. The results are matching with the literature [13-16].

Pseudocode 2: Random sequences generation from ILCG

1: BEGIN

2: Initialise input parameters for ILCG

3: SET number_of_parameters ← 5

4: SET seq_no ← 127000000

5: SET a← 16777214

6: SET pre← 996350000

7: SET c← 12345

8: SET m← 16777215

9: Generate random sequences using the equation (10)

10: Method: genRandomSequences(seq_no, a, pre, c,m)

11: INPUT: Parameters seq_no, a, pre, c,m

12: seq[0] ← pre

13: for i←1 to seq_no do

14: seq[i] = (int) (i * ((a * seq[i - 1] + (i + c))) % m);

15. Return seq

16: end for

17: END

Pseudocode 3: Chaotic sequences generation $x_{sequences}$, $y_{sequences}$ from IL2DCCM

1: BEGIN

2: Initialise parameters for IL2DCCM

3: SET numbers_of_parameters ← 7

4: SET sequ_no ← 127000000

5: SET miu_1←7.100000000000003

6: SET miu_2←1.000000000000003

7: SET gam_1←0.179

8: SET gam_2←0.139

9: SET x_pre>0.500000000000003

10: SET y_pre>0.500000000000004

11: Generate chaotic sequences using the equation (3) & (4)

12: Method: genSequences(sequ_no,miu_1,miu_2, gam_1,gam_2, x_pre, y_pre)

13: INPUT: Chaos parameters sequ_no, miu_1, miu_2, gam_1,gam_2,x_pre, y_pre

14: $x_{sequence}$ [0] ← x_pre;

15: $y_{sequences}$ [0] ← y_pre;

16: for i ← 1 to sequ_no do

$x_{sequence}$[i] = miu_1 * Math.sin($x_{sequence}$ [i-1]) * (1 - Math.sin($x_{sequence}$ [i-1])) + (gam_1 * Math.pow(Math.sin($y_{sequences}$ [i-1]), 2));

$y_{sequence}$[i] = miu_2 * Math.sin($y_{sequences}$ [i-1]) * (1 - Math.sin($y_{sequences}$ [i-1])) + (gam_2 * (Math.pow(Math.sin($x_{sequence}$ [i-1]), 2) + (Math.sin($x_{sequence}$ [i-1]) * Math.sin($y_{sequences}$ [i-1]))));

17: Return $x_{sequences}$

18: Return $y_{sequences}$

19: end for

20: END

TABLE I. RANDOMNESS OF CONFUSION SEQUENCES VALIDATED THROUGH NIST STS 2.1.2.

Statistical Test Name	Proportion	P-Value	Status
Frequency test	10/10	0.739918	Success
Block-Frequency test	10/10	0.534146	Success
Cumulative-Sums-Forward test	10/10	0.991468	Success
Cumulative-Sums-Reverse test	10/10	0.911413	Success
Runs test	9/10	0.122325	Success
Longest-Run test	10/10	0.911413	Success
Rank test	10/10	0.350485	Success
FFT test	10/10	0.911413	Success
Non-Overlapping-Template test	10/10	0.066882	Success
Overlapping-Template test	10/10	0.739918	Success
Universal Maurer's Test	10/10	0.213309	Success
Approximate-Entropy (m=10) test	10/10	0.534146	Success
p-value of Serial test 1	10/10	0.911413	Success
p-value of Serial test 2	10/10	0.350485	Success
Linear-Complexity test 1	10/10	0.350485	Success

TABLE II. RANDOMNESS OF DIFFUSION SEQUENCES VALIDATED THROUGH NIST STS 2.1.2.

Statistical Test Name	Proportion	P-Value	Result
Frequency test	10/10	0.534146	Success
Block-Frequency test	8/10	0.122325	Success
Cumulative-Sums-Forward test	10/10	0.739918	Success
Cumulative-Sums-Reverse test	10/10	0.534146	Success
Runs test	10/10	0.739918	Success
Longest-Run test	10/10	0.739918	Success
Rank test	10/10	0.991468	Success
FFT test	10/10	0.350485	Success
Non-Overlapping-Template test	10/10	0.911413	Success
Overlapping-Template test	9/10	0.534146	Success
Universal Maurer's Test	10/10	0.066882	Success
Approximate-Entropy (m=10) test	10/10	0.122325	Success
p-value of Serial test 1	10/10	0.739918	Success
p-value of Serial test 2	10/10	0.534146	Success
Linear-Complexity test 1	10/10	0.122325	Success

TABLE III. RANDOMNESS OF CONFUSION SEQUENCES VALIDATED THROUGH DIEHARD TESTS.

Statistical Test Name	P-Value	Result
Birthday Spacing test	0.505350	Success
Overlapping 5-Permutation test	0.181724	Success
Binary Rank 31 x 31 Matrices test	0.660460	Success
Binary Rank 32 x 32 Matrices test	0.385863	Success
Binary Rank 06 x 08 Matrices test	0.153396	Success
Bit Stream test	0.675390	Success
Overlapping-Pairs-Sparse-Occupancy test	0.485800	Success
Overlapping-Quadruples-Sparse-Occupancy test	0.468500	Success
DNA test	0.561700	Success
Count the Ones-01 test	0.671861	Success
Parking Lot test	0.515718	Success
Minimum Distance test	0.134781	Success
3D Spheres test	0.282375	Success
Squeeze test	0.458705	Success
Overlapping Sum test	0.932326	Success
Runs-Up test	0.857474	Success
Runs-Down test	0.693456	Success
Craps test	0.957920	Success

TABLE IV. RANDOMNESS OF DIFFUSION SEQUENCES VALIDATED THROUGH DIEHARD TESTS.

Statistical Test Name	P-Value	Result
Birthday Spacing test	0.436200	Success
Overlapping 5-Permutation test	0.561261	Success
Binary Rank 31 x 31 Matrices test	0.720140	Success
Binary Rank 32 x 32 Matrices test	0.953855	Success
Binary Rank 06 x 08 Matrices test	0.454970	Success
Bit Stream test	0.536930	Success
Overlapping-Pairs-Sparse-Occupancy test	0.514700	Success
Overlapping-Quadruples-Sparse-Occupancy test	0.598600	Success
DNA test	0.764800	Success
Count the Ones-01 test	0.595868	Success
Parking Lot test	0.182627	Success
Minimum Distance test	0.378954	Success
3D Spheres test	0.438738	Success
Squeeze test	0.842434	Success
Overlapping Sum test	0.687269	Success

Runs-Up test	0.531294	Success
Runs-Down test	0.928336	Success
Craps test	0.368880	Success

TABLE V. RANDOMNESS OF CONFUSION SEQUENCES VALIDATED THROUGH ENTROPY TEST SUITE.

Statistical Tests	Value	Result
Entropy value	7.999983	Success
Arithmetic Mean value	127.5010	Success
Monte Carlo value	3.142315320	Success
Chi-Square value	307.11	Success
Serial Correlation Coefficient value	-0.000635	Success

TABLE VI. RANDOMNESS OF DIFFUSION SEQUENCES VALIDATED THROUGH ENTROPY TEST SUITE.

Statistical Tests	Value	Result
Entropy value	7.999986	Success
Arithmetic Mean value	127.5150	Success
Monte Carlo value	3.139673	Success
Chi-Square value	254.69	Success
Serial Correlation Coefficient value	0.000585	Success

V. CONCLUSION

This paper effectively involves the improved XSG and LCG, along with improved logistic 2D coupled chaotic map for sequence generation. The efficiency of randomness of the sequences has been studied using the NIST, Diehard and Entropy test suites. The results prove that the confusion-diffusion sequences are cryptographically secure. Hence, the sequences could be employed for encryption 24-bit image pixels. Similarly, other PRNGs can be improved along with any other chaotic equations thereby yields better cryptographic solutions. These secure sequences could be employed in image, audio, video or any file encryption schemes.

ACKNOWLEDGMENT

This study has been carried out in the RUSA funded computer laboratory at Division of Computer and Information Science, Faculty of Science, Annamalai University. Hence, the authors wish to acknowledge the authorities of Annamalai University and RUSA.

REFERENCES

[1] P. Praveenkumar, R. Amirtharajan, K. Thenmozhi, J.B.B. Rayappan, Triple chaotic image scrambling on RGB – a random image encryption approach. Security Comm Netw 8(18), 2015, 3335–3345. http://dx.doi.org/10.1002/sec.1257.

[2] M.Y.M. Parvees, J.A. Samath, B.P. Bose, Confidential storage of medical images–a chaos-based encryption approach, International Journal of Cloud Computing, 7 (1), 2018, 15-39.

[3] M.Y.M. Parvees, J.A. Samath, I K. Raj, B.P. Bose, "A colour byte scrambling technique for efficient image encryption based on combined chaotic map: Image encryption using combined chaotic map", *Proc. Int. Conf. Elect. Electron. Optim. Technol. (ICEEOT)*, pp. 1067-1072, Mar. 2016.

[4] M.Y.M. Parvees, J.A. Samath, B.P. Bose, Audio encryption–a chaos-based data byte scrambling technique, International Journal of Applied Systemic Studies, 8 (1), 2018, 51-75

[5] D. Valli and K. Ganesan, Chaos based video encryption using maps and Ikeda time delay system, The European Physical Journal Plus, 2017, 132:542. https://doi.org/10.1140/epjp/i2017-11819-7.

[6] P. Praveenkumar, R. Amirtharajan, K. Thenmozhi, Medical data sheet in safe havens – A tri-layer cryptic solution. Comp Biol Med 62: 2015, 264-276, http://dx.doi.org/10.1016/j.compbiomed. 2015.04.031.

[7] M.Y.M. Parvees, J.A. Samath, B.P. Bose, The role of improved logistic map in image encryption, *International Conference on Communication &* Security (ICCS-2017), March 3 - 4, 2017 Organized by SASTRA University, Thanjavur, Tamil Nadu, India.

[8] M.Y.M. Parvees, J.A. Samath, B.P. Bose, Medical Images are Safe–an Enhanced Chaotic Scrambling Approach, Journal of medical systems, 41 (10), 2017, 167.

[9] M.Y.M. Parvees, J.A. Samath, B.P. Bose, Secured medical images - a chaotic pixel scrambling approach. Journal of Medical Systems. 40, 232 (2016). http://dx.doi.org/10.1007/s10916-016-0611-5.

[10] A. Rukhin, J. Soto, Nechvatal et al., A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Application, NIST Special Publication 800-22, Revision 1a (Revised: April 2010), Lawrence E. Bassham III, 2010. http://csrc.nist.gov/groups/ST /toolkit/rng/index.html.

[11] Marsaglia, G. DIEHARD: a battery of tests of randomness, 1996. http://www.fsu.edu/pub diehard.

[12] Walker, J.: ENT: A Pseudorandom Number Sequence Test Program, 2008. http://www.fourmilab.ch/random/.

[13] B. Stoyanov, K. Kordov, K. Szczypiorski, Yet Another Pseudorandom Number Generator, International Journal of Electronics and Telecommunications, 63(2), 2017, 195-199. http://dx.doi.org/10.1515 /eletel-2017-0026.

[14] M.Y.M. Parvees, J.A. Samath, B.P. Bose, Cryptographically Secure Diffusion Sequences—An Attempt to Prove Sequences are Random. In: Peter J., Alavi A., Javadi B. (eds) Advances in Big Data and Cloud Computing. Advances in Intelligent Systems and Computing, 750, 2018, 433-442, Springer, Singapore. https://doi.org/10.1007/ 978-981-13-1882-5_37.

[15] M.Y.M. Parvees, J.A. Samath, B.P. Bose, Providing confidentiality for medical image – an enhanced chaotic encryption approach In: Peter J., Alavi A., Javadi B. (eds) Advances in Big Data and Cloud Computing. Advances in Intelligent Systems and Computing, 645, 2018, 309-317, Springer, Singapore. https://doi.org/10.1007/978-981-10-7200-0_28.

[16] M.Y.M. Parvees, J.A. Samath, B.P. Bose, Chaotic Sequences are Cryptographically Secure now – An Improved Chaotic Approach, In Proceedings of International Conference on Innovative Technologies in Electronics, Information and Communication (Intelinc '18), 2018, organized by Department of Electronics and Communication Engineering, Annamalai University,Annamalainagar.