JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Comparative study on TDMRC and E-TDMRC

[1]Antu Annam Thomas

[1]Assistant Professor
[1]Department of Computer Applications,
[1]Mar Thoma College, Tiruvalla, India

Abstract : Among the various encryption techniques Time Dependant Multiple Random Cipher Code (TDRMC) is a novel approach in Character Coding or encryption The random number generator used in TDMRC is Linear Congruential Generator or LCG. Concept of nesting would improve the randomness produced by traditional generator LCG. The mathematical formula for Nested Linear Congruential Generator is same as in traditional formula. But the constant values used will be substituted by another random number generated by a Random Number Generator nested within. By breaking the constant nature the period becomes infinity or no subsequence ever repeats in the generated sequence. By substituting LCG with NLCG the performance of TDMRC is enhanced resulting in Enhanced TDMRC. In this paper TDMRC is compared with E-TDMRC using statistical and graphical methods like Kolmogorov Smirnov Test, Runs Test, Bar Graph Analysis and Scatter Diagram Analysis.

Index Terms **– TDMRC, E-TDMRC, NLCG, LCG**

I. INTRODUCTION

In this E-world security of data is of primary concern. [1] Cryptography is an effective tool for ensuring data security.[2]-[4] Time Dependant Multiple Random Cipher Code is a novel approach in Character Coding or encryption. It follows a symmetric key method and a poly-alphabetic substitution coding system. It uses less complex mathematical operations compared with any other schemes. The technique used for encryption is substitution coding system. TDMRC is 'Mega Extended ASCII Code'. Complexities involved in TDMRC Code are Time Dependency, Poly Alphabetic Nature and use of Pseudo Random Number Generation technique for code generation. Key of TDMRC Code consists of three elements Master Key that is derived from Real Time Clock, Poly Alphabetic Coefficient(PAC), P, that decides the block size and P number of 4 digit subkeys.

Randomness plays a great role in increasing the security offered by cryptographic techniques. [6]-[8]In TDMRC factor of randomness comes in three areas that is finding the Poly alphabetic coefficient- P, generating P Sub keys and generating P Random Series. The random number generator used in TDMRC is a traditional pseudo random number generator known as Linear Congruential Generator or LCG. The major disadvantage of LCG is that the period depends upon the choice of the multiplier, increment and modulus. Only when the combination of values is successful will the series generated be random series with long period. Thus security enhancement of TDMRC can be done by improving the random number generator used or in other words by overcoming the disadvantages of traditional random number generators. [9][10]

The mathematical formula used is same as in traditional formula. But the constant values used, increment and multiplier, will be substituted by another random number generated nested within. By breaking the constant nature the period becomes infinity or no subsequence ever repeats in the generated sequence. Since one random number generator is coming within another it is called nested random number generator. Thus by applying concept of nesting even though a random number in the series repeats there will be no subsequence repeating. Hence the generated series is closer to the true randomness. [11]

In Nested LCG concept of nesting is introduced into traditional LCG. The equation is the same linear piecewise equation as the traditional LCG. But here multiplier and increment is not a constant value but are the random numbers generated by two other random number series. [12][13]

Nesting ensures that sequence is never repeated within the final series being generated and period is infinity. By substituting LCG with NLCG the performance of TDMRC is enhanced and the improved version E-TDMRC comes into play.

II. TDMRC to E-TDMRC

Randomness in TDMRC can be improved by introducing NLCG in place of LCG. Thus E-TDMRC will be using NLCG in place of LCG in all the places where random generation is required.

E-TDMRC's structure includes four levels of complexities. First is the time dependency, this is because any character differs depending upon time. Second is the Poly Alphabetic Nature since the same character at different locations of the plain text is different. Third is the Random sub key values used and the fourth is the random series generated. Both the sub key values and the random series is generated using NLCG in E-TDMRC rather than LCG.

The key of E-TDMRC involves three elements. The key is the main factor that decides the security level of a cryptographic system. Master Key, Poly Alphabetic Coefficient (PAC, P) and P four digit sub keys together forms the key to E-TDMRC.

2.1 Algorithm of TDMRC Code

The Encryption and Decryption algorithm of TDMRC Code is given below. [5]

2.1.1 Encryption Algorithm

Step 1 Decide Poly Alphabetic Coefficient, P. This value decides the number of codes that is to be used simultaneously and is generated by using traditional random number generator LCG.

Step 2 Now P number of sub keys are decided. Sub keys are generated randomly by using LCG. $S_1S_1S_1S_1$, $S_2S_2S_2S_2$,----, $S_pS_pS_pS_p$ are the sub keys that are generated.

Step 3 Read the Real Time Clock Time (System Time) with accuracy to centi second and form an 8 digit number TTTTTTTT. This will act as the Master Key. Since real time clock value is used true randomness is exhibited by the value of master key.

Step 4 Random seed is generated by multiplying the Master Key with the Sub Key values and taking 8 digits of the product from extreme right to form P number of Random Seed values.

Step 5 LCG is employed for generating random series with elements from 0-255, (00000000 - 11111111 in binary) arranged in a random fashion.

Step 6 Now, data is taken in blocks of P number of ASCII characters. ASCII value of each character is read and each character is substituted with element in the random series corresponding to this ASCII value. First random series is used for substitution of first character in block of P ASCII characters, second series is used for substituting second character and so on. The substitution is done using the random series in the cyclic manner. [5]

Flowchart of the Encryption algorithm is given below. Chain of 8 bit ASCII is converted into chain of 8 bit TDMRC through the process of transliteration.

2.1.2 **Decryption Algorithm**

Step 1 Regenerate P number of random seeds and P numbers of random series using the same key used in the Encryption process. The Pseudo Random Number Generation algorithm used is LCG.

Step 2 Now, data is taken in blocks of P number of TDMRC coded characters. Find the ASCII value of each character and then substitute each character with the string character of the serial number value of the element in the random series, the element which is same as the ASCII character in the block. The first character in block of P characters be substituted with the string character of the serial number value of the element from first random series, second character with the string character with the serial number value of the element from second random series and so on.[5]

2.2 **Algorithm of E-TDMRC Code**

The algorithm for Encryption and Decryption of E-TDMRC is given below.

2.2.1 Encryption Algorithm

Step 1 Generate Poly Alphabetic Coefficient, P by using NLCG. PAC decides the number of codes that is to be used simultaneously. PAC is decided by using traditional random number generator LCG.

Step 2 Now P number of Sub Keys are generated using NLCG. Sub keys are generated randomly by using LCG. $S_1S_1S_1S_1$, $S_2S_2S_2S_2$,----, $S_pS_pS_pS_p$ are the sub keys that are generated.

Step 3 Read the Real Time Clock Time (System Time) with accuracy to centi second and form an 8 digit number TTTTTTTT. This will act as the Master Key. Since real time clock value is used true randomness is exhibited by the value of master key.

Step 4 Random seed is generated by multiplying the Master Key with the Sub Key values and taking 8 digits of the product from extreme right to form P number of Random Seed values.

Step 5 NLCG is employed for generating random series with elements from 0-255, (00000000 - 11111111 in binary) arranged in a random fashion.

Step 6 Now, data is taken in blocks of P number of ASCII characters. ASCII value of each character is read and each character is substituted with element in the random series corresponding to this ASCII value. First random series is used for substitution of first character in block of P, ASCII characters and so on. [5]

Flowchart of the Encryption algorithm is given below. Chain of 8 bit ASCII is converted into chain of 8 bit Enhanced TDMRC through the process of transliteration.

2.2.2 Decryption Algorithm

Step 1. Regenerate P number of random seeds and P numbers of random series using the same key used in the Encryption process. The Pseudo Random Number Generation algorithm used is NLCG.

Step 2. Now, data is taken in blocks of P number of E- TDMRC coded characters. Find the ASCII value of each character and then substitute each character with the string character of the serial number value of the element in the random series, the element which is same as the ASCII character in the block. The first character in block of P characters be substituted with the string character of the serial number value of the element from first random series, second character with the string character with the serial number value of the element from second random series and so on.[5]

III. E-TDMRC Vs. TDMRC

Table 3.1 E-TDMRC Vs. TDMRC

E-TDMRC	TDMRC
Random Sub Keys and Poly Alphabetic Coefficient are generated using NLCG	Random Sub Keys and Poly Alphabetic Coefficient are generated using LCG
Since NLCG is used algorithm complexity is more	Algorithm complexity is less since LCG is employed here

Generated key is more random	Generated Key is less random
Random Series used for character coding is generated using NLCG	Random Series used for character coding is generated using NLCG
Cryptographically secure and more effective encryption and decryption algorithm	Cryptographically secure but less effective cryptographic algorithm than Enhanced TDMRC
Cryptanalysis not possible	Cryptanalysis not possible
Computation time required is more than TDMRC	Computation time require is negligible

IV. Statistical Analysis

4.1 Kolmogorov Smirnov Test

The Kolmogorov-Smirnov test is a non-parametric test. It is 'goodness of fit' test to test the quality of Random Number Generators. KS Test is conducted on both NLCG and LCG results and the analysis table is given below. The test hypothesis is given below:

H_0 = Sequence being tested is random
H_a = Sequence being tested is not random
K^+ = maximum observed deviation below the expected cdf $(\max(j/n - X_j))$
K^- = minimum observed deviation below the expected cdf$(\max(X_j - (j - 1)/n)$

Considering the case of TDMRC
$K^+ = 0.177864$
$K^- = 0.054947$
From KS test table (Table 4.1) at n=30 and 1-α=0.9
K=0.21756
$K^+ < K$ and $K^- < K$ hence sequence generated by LCG in TDMRC is random and pass KS test

Considering the case of NLCG:
$K^+ = 0.0994792$
$K^- = 0.0611979$
From KS test table (Table 4.2) at n=30 and 1-α=0.9
K = 0.21756

$K^+ < K$ and $K^- < K$ hence sequence generated by NLCG is random and pass KS test

Table 4.1 K S Test analysis of Random Series generated by LCG in TDMRC

J	Random number in sorted order X_j	Normalized X_j / 256	$(j/n-X_j)$	$X_j-(j-1)/n$
1	1	0.00390625	0.029427083	0.00390625
2	13	0.05078125	0.015885417	0.017447917
3	16	0.0625	0.0375	-0.00416667
4	28	0.109375	0.023958333	0.009375
5	29	0.11328125	0.053385417	-0.02005208
6	41	0.16015625	0.03984375	-0.00651042
7	42	0.1640625	0.069270833	-0.0359375
8	57	0.22265625	0.044010417	-0.01067708
9	65	0.25390625	0.04609375	-0.01276042
10	66	0.2578125	0.075520833	-0.0421875
11	79	0.30859375	0.058072917	-0.02473958
12	83	0.32421875	0.07578125	-0.04244792
13	87	0.33984375	0.093489583	-0.06015625
14	88	0.34375	0.122916667	-0.08958333
15	89	0.34765625	0.15234375	-0.11901042
16	91	0.35546875	0.177864583	-0.14453125
17	105	0.41015625	0.156510417	-0.12317708
18	109	0.42578125	0.17421875	-0.14088542
19	144	0.5625	0.070833333	-0.0375
20	149	0.58203125	0.084635417	-0.05130208
21	155	0.60546875	0.09453125	-0.06119792
22	165	0.64453125	0.088802083	-0.05546875
23	176	0.6875	0.079166667	-0.04583333
24	193	0.75390625	0.04609375	-0.01276042
25	201	0.78515625	0.048177083	-0.01484375
26	209	0.81640625	0.050260417	-0.01692708
27	210	0.8203125	0.0796875	-0.04635417
28	242	0.9453125	-0.01197917	0.0453125
29	253	0.98828125	-0.02161458	0.054947917
30	254	0.9921875	0.0078125	0.025520833

Table 4.2 K S Test analysis of Random Series generated by NLCG in E-TDMRC

J	Random numbers in sorted order X_j	Normalized X_j / 256	$(j/n - X_j)$	$X_j - (j-1)/n$
1	7	0.02734375	0.0059896	0.0273438
2	11	0.04296875	0.0236979	0.0096354
3	30	0.1171875	-0.0171875	0.0505208
4	33	0.12890625	0.0044271	0.0289063
5	37	0.14453125	0.0221354	0.0111979
6	40	0.15625	0.04375	-0.0104167
7	44	0.171875	0.0614583	-0.028125
8	59	0.23046875	0.0361979	-0.0028646
9	63	0.24609375	0.0539063	-0.0205729
10	82	0.3203125	0.0130208	0.0203125
11	85	0.33203125	0.0346354	-0.0013021
12	89	0.34765625	0.0523438	-0.0190104
13	92	0.359375	0.0739583	-0.040625
14	94	0.3671875	0.0994792	-0.0661458
15	111	0.43359375	0.0664063	-0.0330729
16	125	0.48828125	0.0450521	-0.0117188
17	132	0.515625	0.0510417	-0.0177083
18	142	0.5546875	0.0453125	-0.0119792
19	146	0.5703125	0.0630208	-0.0296875
20	151	0.58984375	0.0768229	-0.0434896
21	177	0.69140625	0.0085937	0.0247396
22	184	0.71875	0.0145833	0.01875
23	194	0.7578125	0.0088542	0.0244792
24	196	0.765625	0.034375	-0.0010417
25	203	0.79296875	0.0403646	-0.0070313
26	229	0.89453125	-0.0278646	0.0611979
27	234	0.9140625	-0.0140625	0.0473958
28	241	0.94140625	-0.0080729	0.0414063
29	244	0.953125	0.0135417	0.0197917
30	248	0.96875	0.03125	0.0020833

From the above analysis it is clear that both LCG and NLCG generated series pass the KS Test.

4.2 Runs Test

A series of increasing or decreasing values is called a run. Number of increasing or decreasing values defines the length of the run. For starting the runs test median of first thirty elements are found out. If a value in the series is less than the median then it is denoted by -1 otherwise +1. After forming the series of +1 and -1, runs of +1 and -1 are counted and hypothesis testing is done.

H_0: Sequence is random

H_a : Sequence is not random

From the sequence generated by LCG, 30 samples are taken and median is calculated. Median is got as 89. Now all the values greater than 89 is denoted as +1 and values less than 89 as -1. From the analysis shown below(Table 4.3), number of runs is got as 15. Now, n_1, number of -1, is 16 and n_2 , number of +1 is 14. From runs table the test is passed if the number of runs is between 10 and 22. Here number of runs is 15 and the generated sequence is thus concluded to be random.

From the sequence generated by NLCG, 30 samples are taken and median is calculated. Median is got as 118. Now all the values greater than 118 is denoted as +1 and values less than 118 as -1. (Table 4.4)Number of runs is got as 14. Now, n_1, number of -1, is 15 and n_2 , number of +1 is 15. From runs table the test is passed if the number of runs is between 10 and 22. Here number of runs is 14 and the NLCG generated sequence also posses the property of randomness.

Table 4.3 Runs Test analysis of Random Series generated by LCG in TDMRC

Random Series	Value>median +1 else -1
	Median =89
144	1
28	-1
13	-1
176	1
209	1
253	1
193	1
16	-1
254	1
242	1
79	-1
57	-1
109	1
42	-1
87	-1
83	-1
201	1
89	1
66	-1
105	1
65	-1
1	-1
91	1
41	-1
29	-1
88	-1
210	1
149	1
165	1
155	1

Table 4.4 Runs Test analysis of Random Series generated by NLCG in Enhanced TDMRC

Random Series NLCG	Value>median +1 else -1
	Median =118
92	-1
89	-1
82	-1
111	-1
44	-1
229	1
194	1
63	-1
184	1
85	-1
146	1
203	1
40	-1
37	-1
30	-1
59	-1
248	1
177	1
142	1
11	-1
132	1
33	-1
94	-1
151	1
244	1
241	1
234	1
7	-1
196	1
125	1

V. CRYPTANALYSIS OF E-TDMRC

Depending upon the level of information a cryptanalyst have, cryptanalysis can be of three types.

i. Ciphertext Only Attack – Here the attack is solely based on the Ciphertext which is the only information that a Cryptanalyst have
ii. Known Plaintext Attack – Here the cryptanalyst have a Plaintext and its corresponding Ciphertext as his information.
iii. Chosen Plaintext Attack – Here the attack is based on a Chosen Plaintext and its corresponding Ciphertext.

In the first case the attack is not possible because Exhaustive Search Method is not practically possible in case of E-TDMRC. At any instant Ciphertext will have a combined complexity of Random Seed Values which are generated based on Real Time Clock value and a randomly chosen Pixel Value, Poly Alphabetic Coefficient, Random Number Generation Technique and Nesting. [5]

Quantitatively in TDMRC as already mentioned Master key is an 8 digit number derived from the Real Time Clock Value with 8640000 possible values. As per the algorithm the first seed value is obtained by multiplying the first Sub Key Value with Master Key Value. Sub Key Value is having 1000 possible values. Thus the possible number of seed value for first code generation is 864×10^7. Similarly there are 864×10^7 possible values for each code generation. [5]

In the case of E-TDMRC in addition to the real time clock value, intensity value of a pixel randomly chosen from the image is also considered. Thus the possible number of seed values for each code generation is $864 \times 10^7 \times 256^t$, where t is the number of pixels in the image and each pixel have 256 possible intensity values.

In the case of Enhanced TDMRC Poly Alphabetic Coefficient is also randomly chosen.

Let it be P then the possible number of keys are $(864 \times 10^7 \times 256^t)^P$.

Thus from the above discussion it is clear that cryptanalysis is not possible in Ciphertext only situation.

Poly Alphabetic and Random nature of E-TDMRC ensures Brute Force Attack also is not possible.

The other two classifications of Cryptanalysis Known Plaintext Attack and Chosen Plaintext Attack will not be a threat to E-TDMRC. This is because the Ciphertext generated for the same Plaintext will be different at different instant due to the Poly Alphabetic Nature of E-TDMRC.

E-TDMRC is thus cryptographically very secure and cryptanalysis is rather impossible.

VI. COMPUTATIONAL TIME ANALYSIS

Both TDMRC and E-TDMRC follow the same algorithm except for the choice of random number generator. Thus the computation time difference between TDMRC and E-TDMRC is just because of difference in random number generators used. Computational time is little more for E-TDMRC since NLCG is used. When the computation time is negligible for TDMRC, it is about 0.0012 seconds more for E-TDMRC. Comparison of computational time for E-TDMRC and TDMRC is given below.

From the figure given below (Figure 6.1)it can be seen that computational time is less for TDMRC. But it is only about 1.2 milliseconds for E-TDMRC. This increase in computation time may be counted negligible when its advantages are taken into account.

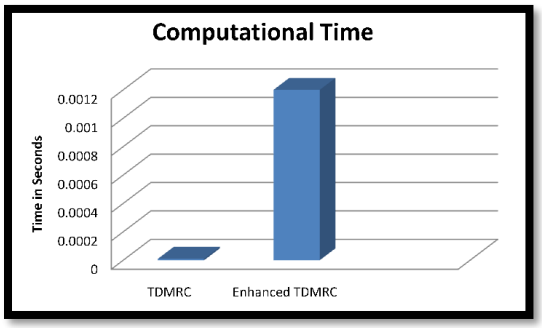

Figure 6.1 Computational Time Analysis

VII. GRAPHICAL ANALYSIS

7.1 Bar Graph Analysis

Bar graph was plotted for the both the random series. In the case of TDMRC the random generator used is LCG and the period of the generated series depends upon the value of the constants chosen. While in the case of E-TDMRC the generated series has period infinity. Thus in the case of TDMRC extra care should be taken for the series to contain all numbers from 0 to 255 without any repetition. Period is infinity for NLCG because the increment and the multiplicand value is never constant it forms the random number generated by a nested random series.

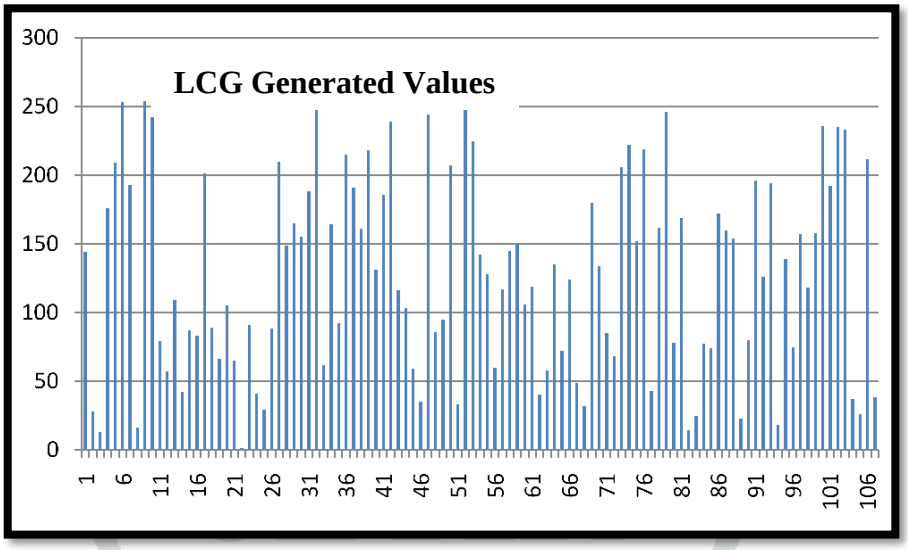

Figure 7.1 Bar Graph Analysis of Random Series generated by LCG

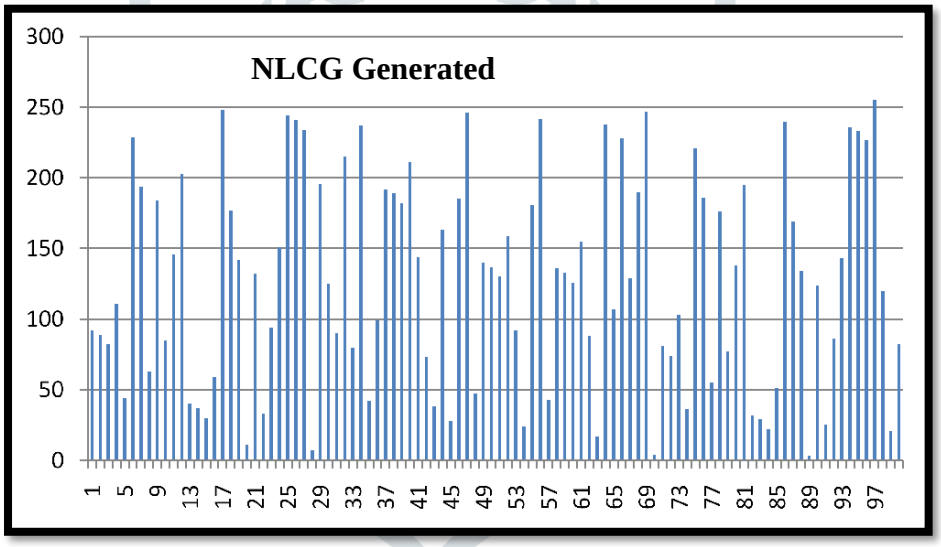

Figure 7.2 Bar Graph Analysis of Random Series generated by NLCG

5.2 Scatter Diagram Analysis

For analyzing first 30 elements are chosen from both the NLCG and LCG series. Points on the graph are divided into four quadrants. If there are X points on the graph, Count X/2 points from top to bottom and draw a horizontal line. Count X/2 points from left to right and draw a vertical line. Here 30 points are considered so lines are drawn after 15 points and graph divided into four quadrants.

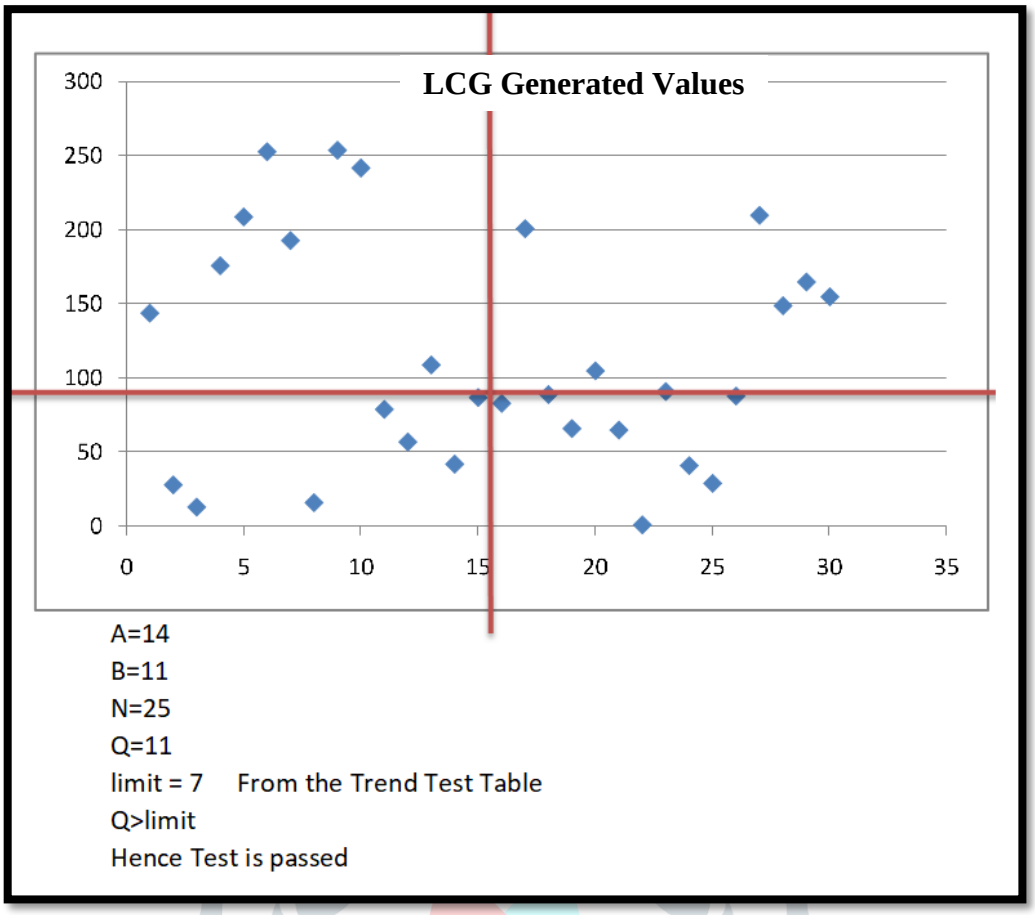

Figure 7.3 Scatter Diagram Analysis of LCG generated Series

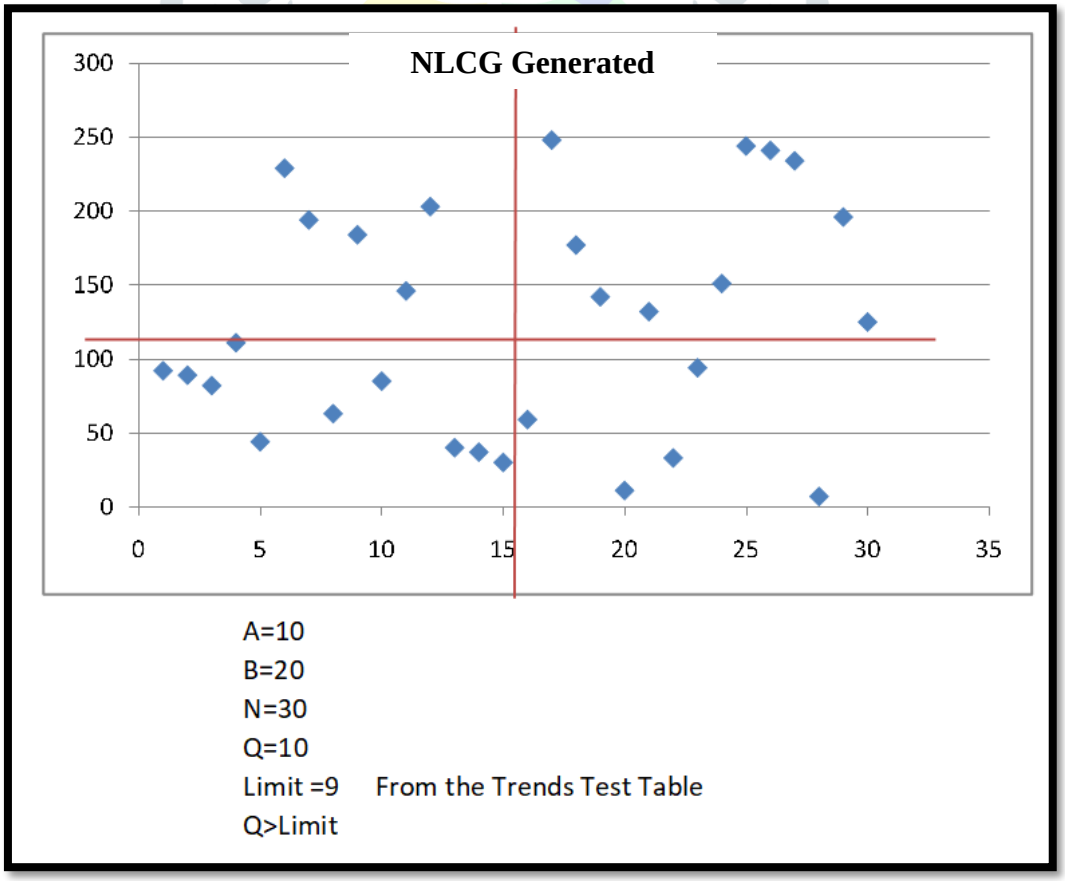

Figure 7.4 Scatter Diagram Analysis of NLCG generated Series

Scatter Graph analysis is conducted for both TDMRC and E-TDMRC generated Series.

A = points in upper left + points in lower right
B = points in upper right + points in lower left
Q = the smaller of A and B
N = A + B
If Q is less than the limit, the two variables are related.
If Q is greater than or equal to the limit, the pattern could have occurred from random chance.

Here, in the case of TDMRC generated series

A=14, B=11, N=25, Q=11
From the Runs Test Table, limit=7
Now, Q>limit
Hence the test is passed.

Now, consider series generated by E-TDMRC

A=10, B=20, N=30, Q=10
limit=9, Q>limit
Hence the test is passed

It is proved from the scatter plot analysis that the elements in both the series has no correlation. Further, Scatter diagram analysis proves that the series posses good randomness and that there is no linear relationship or correlation between the generated random values. Scatter diagram analysis proves that the values generated in both the series are by random chance.

VIII. CONCLUSION

Performance of TDMRC and E-TDMRC was compared using statistical and graphical methods. By substituting LCG with NLCG performance improvement of TDMRC was made possible. LCG as such is not considered good for security related applications. Nesting makes LCG suitable for security applications. The statistical and graphical analysis also the proved the improved performance of E-TDMRC. Thus it can be concluded that E-TDMRC is more apt for cryptographic and other security related activities when compared to TDMRC.

References

[1]. Harshavardhan Kayarkar, Sugata Sanyal, "Classification of Various Security Techniques in Databases and their comparative analysis", ACTA TECHNICA CORVINIENSIS –Bulletin of Engineering, Fascicule 2 [April–June],2012,pp. 135-138
[2]. The Art of Computer Programming - Donald E Knuth – Vol 2- Addison Wesley Publication – 1968
[3]. Introduction to Cryptography with Coding Theory – Wade Trappe and Lawrence C. Washington
[4]. *Cryptography* and Network Security Principles and Practices, Fourth Edition. By *William Stallings*. Publisher: Prentice Hall. Pub Date: November 16, 2005.
[5]. "Data Security in Fault Tolerant Hard Real Time Systems, Use of Time Dependant Multiple Random Cipher Code", A thesis submitted by Varghese Paul in partial fulfillment of the requirements for the degree of Doctor Of Philosophy of Cochin University of Science and Technology
[6]. Kinga Marton, Alin Suciu, Losif Ignat, "Randomness in Digital Cryptography: A Survey" Romanian Journal Of Information Science and Technology ,Volume 13, Number 3, 2010, 219–240
[7]. Norm Matloff, "Random Number Generation",February 21, 2006
[8]. Harald Niederreiter, "Random Number Generation and QuasimMonte Carlo Methods", Society for Industrial and Applied Mathematics, Philadelphia, 1992.
[9]. "Linear Congruential Generators" by Joe Bolte, Wolfram Demonstrations Project.
[10]. "True Random Number Generators Secure in a Changing Environment", Boaz Barak, Ronen Shaltiel, and Eran Tromer, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science , Rehovot, ISRAEL
[11]. Adi A. Maaita, Hamza A. A. Al_Sewadi, "Deterministic Random Number Generator Algorithm for Cryptosystem Keys", International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:9, No:4, pp 972-977, 2015
[12]. Antu Annam Thomas and Varghese Paul, "Performance Enhancement of Cryptographic Algorithms by Increasing Randomness through Nesting In Random Number Generators", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 2, Issue 5, May 2017, pp. 203- 209
[13]. Antu Annam Thomas and Varghese Paul, "Nested Random Number Generator", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 7, Issue 5, May 2017, pp.767-773