ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, October 2017

Enhancement of Security in Cryptographic Systems by Improving Randomness

[1] Antu Annam Thomas, [2] Varghese Paul
[1] Dept. of Computer Application Mar Thoma College, Thiruvalla
[2] Dept. of Information Technology Rajagiri School of Engineering and Technology Rajagiri Valley, Kakkanad

***Abstract*— Randomness plays an important role in many of the applications where natural selection process needs to be simulated. The term randomness also finds an important and irreplaceable role in the area of data security and cryptography. There are various traditional random number generators. Multiplicative Congruential Generator (MCG) is one among them. The randomness of the generated sequence is enhanced by the concept of nesting introduced into the algorithm. In this work the traditional MCG is compared with Nested Multiplicative Congruential Generator(NMCG) using statistical and graphical analysis techniques. The most important advantage of NMCG is that the period of the generated sequence is infinity. In the case of MCG the period depends upon the value of multiplicand. The analysis performed also concludes that NMCG is a better random number generator when compared to MC.**

***Index Terms*— Cryptography; Data Security; True Random Number Generator; Pseudo random number generator; Multiplicative Congruential Generator; Prime number; Kolmogorov Smirnov Test; Runs Test**

I. INTRODUCTION

Random Numbers have a lot of applications where unpredictability is desired lucky draw, natural phenomena simulation, cryptography and so on. A series is said to be random if it is unpredictable, independent and uniform. Random numbers are generated by Random Number Generators. The efficiency of the generator depends on how much unpredictable, independent and uniform is the generated series.[1][2][3]

There are two types of random number generators true random number generator and pseudo random number generator. While true random number generators use real world phenomena for generation, pseudo random number generator uses computational algorithms for generation.[8]Though there are many true random number generators such as Hotbits, Laser, Random.org and so on, pseudo random number generators are used in most of the applications where randomness comes into play. [6][7][19] For better performance Pseudo Random number generators and true random number generators are used in combination also.[14][15][9] In this paper two random number generators Multiplicative Congruential Generator is compared with Nested Multiplicative Random Number Generator.[10][11]The analysis concludes that nested concept when introduced the traditional MCG has improved its performance. In nested random series the period is always infinity that is a subsequence never repeats.[12]A better random number generator promises betterment of cryptographic systems.[13]

II. MULTIPLICATIVE CONGRUENTIAL GENERATOR (MCG)

Multiplicative Congruential Generator is one of the oldest and most popular random number generator. MCG is simple and easy to implement it is a variation of traditional LCG.

Random series is generated based on a equation given below.

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

Here,

X is the sequence of random numbers
m, $0<m$ - modulus
a, $0<a<m$ - multiplier
X_0, $0 \leq X_0 < m$ - seed value

MCG is fast and requires very less memory. Period of the generated series depends upon the value of 'm'. MCG is not suitable for applications like cryptography where high security is demanded because the series repeats itself after the period length.[16][17].

III. NESTED MULTIPLICATIVE CONGRUENTIAL GENERATOR(NMCG)

In Nested MCG concept of nesting is introduced into traditional MCG. The series is generated based on the equation given below. The equation is the same as that of traditional MCG. But here multiplier is not a constant value as 'a' in equation (1). 'multi' is a random number generated by another random number series.

NLCG consist of three steps:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, October 2017

i. Getting the seed value
ii. Generating the series

Getting the seed value:

Based on the current system clock value read a pixel value of the current picture captured by system camera is read. Based on the pixel value read two prime numbers $p1_0$ and $p2_0$ are generated. Here $p1_0$ is the greatest prime number less than the read pixel value and $p2_0$ is the smallest prime number greater than the read pixel value.

Seed value is given by,

$$X_0=p1_0*p2_0 \bmod m \qquad (2)$$

m is relatively prime to $p1_0*p2_0$

Thus seed value X_0 is the product of two prime numbers. Generated seed value is cryptographically secure due to two factors, difficulty in factorizing product of two prime numbers and true randomness introduced, clock value and pixel value. The above mentioned complexity increases the efficiency of the system and makes the job of cryptanalyst difficult or rather impossible.[20]

Generating the series

$$X_i=(X_{i-1}*b_i)\bmod m \qquad (3)$$

$$b_i=f_i*p1_{i-1}*p2_{i-1} \qquad (4)$$

f_i is the pixel value read from the image based on $p1_{i-1}$ and $p2_{i-1}$

Now based on $p1_{i-1}$ and $p2_{i-1}$ next pair of prime numbers is generated $p1_i$ and $p2_i$. Equations (3) and (4) together generate the random number series. The next element in the generated random sequence 'X_i' not only depends on previous value X_{i-1} but also on b_i which are next element of random series generated by the equations (4). Thus another random series values contribute for final random series generation. That is one random series is nested within another series. Nesting ensures that sequence is never repeated within the final series being generated and period is infinity. NMCG algorithm is pictorially represented in the flowchart given below.

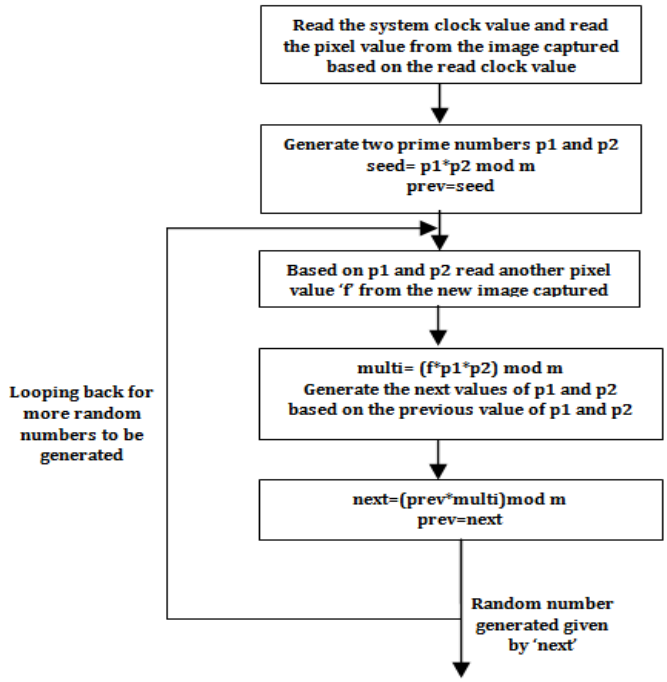

FIG. 1. Flowchart Of Nmcg

NMCG is cost effective since no expensive hardware is required. Seed value generated is sufficiently complex since true randomness and prime factorization problem is used in this step. Nested concept used during series generation makes the series unpredictable and random and subsequence never repeats in the generated sequence. [6][15] Complexities involved in this method include true entropy sources introduced in the generation, prime factorization problem, concept of nesting used in the algorithm.

IV. IMPLEMENTATION

Both mcg and nmcg were implemented in matlab and output was analyzed. System clock value was read and a pixel value was read from the current image captured based on the pixel value read. Then two prime numbers $p1_0$ and $p2_0$ was generated. Seed value is evaluated based on equation (2). The same seed value was used for mcg also.

Now the series is generated for nmcg using the equations (3) and (4) and for mcg using the equation (1). The output got for mcg and nmcg after implementing the algorithm is given below. 'm' was chosen to be 197.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, October 2017

```
Columns 1 through 15

  86   107   186    82     1    84    10    96    61    93    29   114    25   108    73

Columns 16 through 30

  37   183    25   129    71   172   124   153    48    89    89    86    63   169   162

Columns 31 through 45

  98   126    69     6   147    96   131   114   151   131   175   101    95     9   156

Columns 46 through 60

  30   173   136   101   135    95   147    68    14    45    21   146   193    61    87

Columns 61 through 75

  44    80   143   171   194    87    67   111    85   114    90    37    35   176    83

Columns 76 through 90

 146    40    23    81    11   119    38     1    14    54    75   155   102   146   100

Columns 91 through 105

  75   175   181   157    13   190   131    45    93    11     8   123   112     2     5

Columns 106 through 120

 171    14   168    61   165   142   195   148    44   101    78    19   191    97    12
```

FIG. 1. Output Showing The First 105 Elements In The Random Series Generated By Mcg

```
Columns 1 through 15

  71    89    20   186    75     8    35    30    82    14    12   151    45   123    21

Columns 16 through 30

  18   128   166    86   130    27   192    52   129   195   139    91    78    95   194

Columns 31 through 45

 110    38   117    44    94   165    57    77    66   141   149   184    17    99   113

Columns 46 through 60

 125    79   124    50    71    89    20   186    75     8    35    30    82    14    12

Columns 61 through 75

 151    45   123    21    18   128   166    86   130    27   192    52   129   195   139

Columns 76 through 90

  91    78    95   194   110    38   117    44    94   165    57    77    66   141   149

Columns 91 through 105

 184    17    99   113   125    79   124    50    71    89    20   186    75     8    35
```

FIG. 2. Output showing the first 105 elements in the random series generated by nmcg

V. RESULT ANALYSIS

1) scatter diagram

Scatter diagram analysis proves that the series generated by nmcg is better when compared to mcg.

For analyzing first 30 elements are chosen. 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.

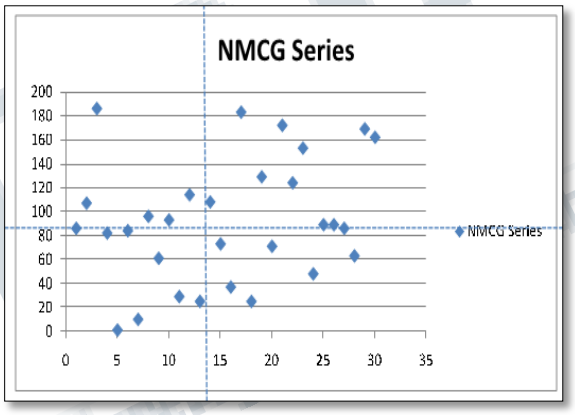

Fig. 4. Nmcg scatter diagram

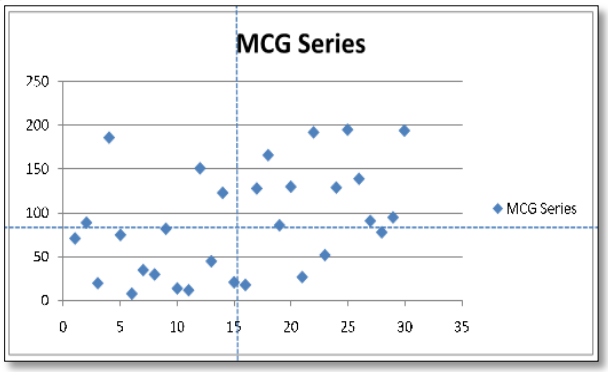

Fig. 5. Mcg scatter diagram

Now,

A = points in upper left + lower right (5)

B = points in upper right + lower left (6)

Q = smaller of a and b (7)

N=a+b (8)

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, October 2017

equations (5), (6), (7) and (8) is considered and the q value is compared with the limit value read from the trend test table. Limit given in the trend test table for sample of 30 is 9. Here for nmcg value of a=12 and b=18 q is 12 and is greater than the limit 9 hence the numbers in the series is drawn by random chance. But for mcg value of a=8 and b=22 q is 8 which is less than the limit value 9 hence numbers in the series are somehow related. Hence scatter diagram analysis shows that nmcg is a better random number generator than mcg.

2) bar graph analysis

the bar graph is plotted for first 200 random numbers generated both for mcg and nmcg.

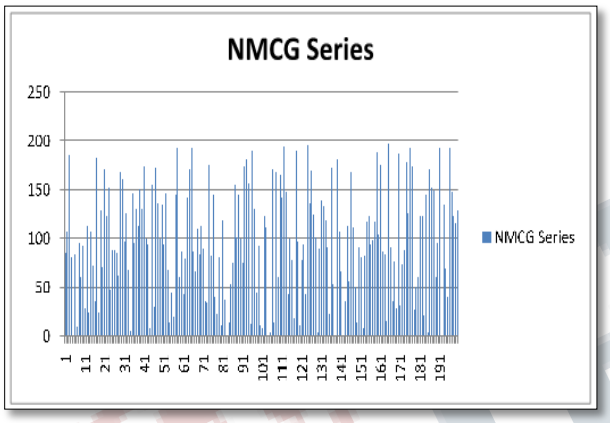

Fig. 6. Bar graph of nmcg output

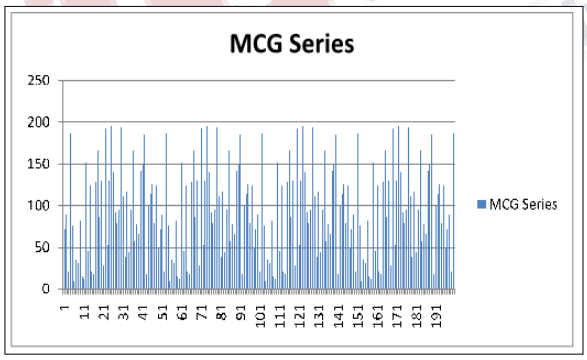

Fig. 7. Bar graph of mcg ouput

the plotted bar graph shows that the generated sequence is random and a subsequence never repeats for nmcg. But the case is different with mcg here the period is 49 that is subsequence repeats after every 49 numbers being. In the bar graph generated by mcg the same pattern repeats after every 49 numbers plotted. But in the case of nmcg the subsequence never repeats. This is because the multiplicand value is never constant it forms the random number in another nested random series. Hence bar graph analysis also shows nmcg is better than mcg as a random number generator.

3) kolmogorov smirnov test

Ks test can be used to check the randomness of the numbers generated by a rng that is allowed to take on any value within a certain interval, leading to a continuous cdf [16]. Ks test is conducted on both nmcg and mcg results and the analysis table is given below.[18]

H0 = sequence being tested is random

Ha = sequence being tested is not random

Table i. K s test analysis of nmcg

j	random	normalised	(j/n-xj)	xj-(j-1)/n
1	1	0.005076142	0.02825719	0.00507614
2	10	0.050761421	0.01590525	0.01742809
3	25	0.126903553	-0.0269036	0.06023689
4	25	0.126903553	0.00642978	0.02690355
5	29	0.147208122	0.01945854	0.01387479
6	37	0.187817259	0.01218274	0.02115059
7	48	0.243654822	-0.0103215	0.04365482
8	61	0.30964467	-0.042978	0.07631134
9	63	0.319796954	-0.019797	0.05313029
10	71	0.360406091	-0.0270728	0.06040609
11	73	0.370558376	-0.0038917	0.03722504
12	82	0.416243655	-0.0162437	0.04957699
13	84	0.426395939	0.00693739	0.02639594
14	86	0.436548223	0.03011844	0.00321489
15	86	0.436548223	0.06345178	-0.0301184
16	89	0.45177665	0.08155668	-0.0482234
17	89	0.45177665	0.11489002	-0.0815567
18	93	0.472081218	0.12791878	-0.0945854
19	96	0.487309645	0.14602369	-0.1126904
20	107	0.543147208	0.12351946	-0.0901861
21	108	0.54822335	0.15177665	-0.1184433
22	114	0.578680203	0.15465313	-0.1213198
23	124	0.629441624	0.13722504	-0.1038917
24	129	0.654822335	0.14517766	-0.1118443
25	153	0.776649746	0.05668359	-0.0233503
26	162	0.822335025	0.04433164	-0.0109983
27	169	0.85786802	0.04213198	-0.0087986
28	172	0.873096447	0.06023689	-0.0269036
29	183	0.92893401	0.03773266	-0.0043993
30	186	0.944162437	0.05583756	-0.0225042

From the table generated for nmcg output

k+ = 0.1546531

k- = 0.0763113

From ks test table at n=30 and 1-α=0.9

k=0.21756

K+ < k and k- < k hence sequence generated by nmcg is random and pass ks test

Table ii. K s test analysis of mcg

From the table generated for mcg output

K+ = 0.1510998

K- = 0.0774958

From ks test table at n=30 and 1-α=0.9

k=0.21756

K+ < k and k- < k hence sequence generated by mcg is random and pass ks test

The output analysis shows that the sequence generated by

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, October 2017

both mcg and nmcg are random.

4) Runs Test

Run can be defined as a series of increasing values or a series of decreasing values. Length of the run is the number of increasing, or decreasing, values. For starting the runs test median of first thirty elements are found out. If a value in the series is less than median then it is denoted by -1 otherwise +1. Now runs are counted for this series of +1 and -1 and hypothesis testing is done.

Ho : sequence is random

Ha : sequence is not random

From the sequence generated by mcg 30 samples are taken and median is calculated. Median is got as 84. Now all the values greater than 84 is denoted as +1 and values less than 84 as -1. Number of runs is got as 18. Now, n1, number of -1, is 14 and n2 , number of +1 is 16. From runs table the test is passed if the number of runs is between 10 and 22. Here number of runs is 18 and hence the generated sequence is not random. For nmcg median is 87.5, n1 is 15 and n2 is 15. Number of runs is 20 which is between 10 and 22, hence the generated sequence is random. Thus the runs test also proves that both nmcg and mcg are good random number generators.

VI. CONCLUSION

In this paper Nested Multiplicative Congruential Generator (NMCG) is compared with Multiplicative Congruential Generator (MCG). True random source, prime factorization problem and nesting all together contributes to the enhanced behavior of NMCG. For NMCG the period of the generated sequence is always infinity but for MCG the period depends upon the value of multiplicand chosen. Period is infinity for NMCG since the value of multiplicand is never constant. For applications that require high degree of randomness the period needs to be infinity that is a subsequence should never repeat. The presence of period makes the work of cryptanalyst easier. Statistical and Graphical analysis conducted on the generated sequence also proves NMCG to be good random number generator.

REFERENCES

[1] B. Schneier, “Applied cryptography: protocols, algorithms, and source code in C,” Second Edition, John Wiley & Sons, 1996.

[2] D. Dilli, Madhu S., “Design of a New Cryptography Algorithm using Reseeding -Mixing Pseudo Random Number Generator,” IJITEE, vol.52, No. 5, 2013

[3] K. Marton, A. Suciu, C. Sacarea, and Octavian Cret, “Generation and Testing of Random Numbers for Cryptographic Applications,” Proceedings of the Ramanian Academy, Series A, Vol. 13, No. 4, 2012, PP 368–377.

[4] Wikipedia, “Pseudorandom number generator”, Last visited December 2014.

[5] D. Dilli, and S. Madhu, “Design of a New Cryptography Algorithm using Reseeding -Mixing Pseudo Random Number Generator,” IJITEE, vol. 52, no. 5, 2013.

[6] “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

[7] David DiCarlo, “Random Number Generation: Types and Techniques,” A Senior Thesis submitted in partial fulfillment of the requirements for graduation in the Honors Program Liberty University Spring 2012.

[8] McNichol, Tom (2003-08-11). "Totally Random". Conde Nast Publications. p. 2. Retrieved 2009-10-23. Mads Haahr, a lecturer in computer science at Trinity College in Dublin, designed the system

[9] T. Simul, S.M. Assad, P.K. Lam “Real time demonstration of high bitrate quantum random number generation with coherent laser light”, Appl Phys Lett 98:231103-1-3

[10] Atsushi Uchida, Kazuya Amano, Masaki Inoue, Kunihito Hirano, Sunao Naito, Hiroyuki Someya, Isao Oowada, Takayuki Kurashige, Masaru Shiki, Shigeru Yoshimori, Kazuyuki Yoshimura & Peter Davis, “Fast physical random bit generation with chaotic semiconductor lasers”, Nature Photonics 2, 728 - 732 (2008)

[11] Sunar, B., Martin, W.J., Stinson, D.R. “A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks”, Computers, IEEE Transactions on (Volume:56 , Issue: 1), Jan. 2007, pp. 109 – 119

[12] Hamed Rahimov, Majid Babaie, Hassan Hassanabadi, "Improving Middle Square Method RNG Using Chaotic Map", Applied Mathematics, 2011, 2, 482-486

[13] Chan, H. "Random number generation". Retrieved 10/16/2011 from http://fuchun00.dyndns.org/~mcmintro/random.pdf, 2009.

[14] Nishimura, T, "Tables of 64-bit mersenne twisters" , ACM Transactions on Modeling and Computer Simulation, 10(4), 348-357, 2000.

[15] 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

[16] D.Y. Dowham and F.D.K. Roberts,"Multiplicative Congruential Pseudo Random Number Generators"

[17] Donald E. Knuth (6 May 2014). Art of Computer Programming, Volume 2: Seminumerical Algorithms. Addison-Wesley Professional. pp. 4–. ISBN 978-0-321-63576-1.

[18] "Testing Random Number Generators", Dan Biebighauser University of Minnesota - Twin Cities REU Summer 2000

[19] Antu Annam Thomas and Varghese Paul, "Random Number Geneeration Methods a Survey", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 1, January 2016, pp.556-559

[20] 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