Generating Random Data using 3 Nonlinear Functions

Sharaf A. Alhomdy, PhD
Associte Prof.
Dept. of IT, Faculty of Computer and Information Technology, Sana'a University, Yemen.

Saleh N. Abdullah, PhD
Associate Prof.
, Dept. of Comp. Science, Khawlan College, Sana'a University, Yemen.

Malek N. Algabri, PhD
Asst. Prof.,
Dept. of Comp. Science, Faculty of Computer and Information Technology, Sana'a University, Yemen

ABSTRACT
In recent years, security systems are built on increasingly strong cryptographic algorithms. Hence, most of the applications which require a high level of security system should include the random number generators (RNGs). Furthermore, most RNGs use nonlinear functions to generate secret quantities for protecting the information that need high level of the security for these applications. Unfortunately, the main deficiencies in the available RNGs are the short period of its repeat cycle length and also the predefined values determined by the static factors. Therefore, this paper describes a new technique to generate random data using 3 nonlinear functions which will extend the periodic cycle length of the repetition that enhances the system security.

Keywords
Seed; Period; Static or Dynamic Factors; RNG; Security; Nonlinear Function

1. INTRODUCTION
Recently, a lot of developments have taken place in security systems using RNGs. The RNGs are nothing more than deterministic algorithms that produce numbers with certain distributional properties. In general, the idea behind a good generator supposes that it is not computationally easy to distinguish the output of the generator from truly random numbers, if the seed is not known. Commonly, there are many techniques of RNGs; some of them can use the linear and the others have used nonlinear functions that are used to create a good cryptographic security for the system [1], [2], [3], [4], [5]. For doing so, those techniques have been used by many applications to generate secret quantities for protecting the information that need high level of the security. For instance, the data generated for credit cards, bank account numbers, etc.

Therefore, in the liner RNGs functions the sophisticated attacker of these security systems may find it easier to reproduce the environment to locate secret quantities in the whole of the number space. Whereas, on the other hand, the nonlinear RNGs are useful during the process of creating the randomized data. Commonly, in these functions it is difficult to detect generate random number if the seed changes periodically. Also it is difficult to detect the next generated random number based on the first one. Hence, indeed the nonlinear function is the cornerstone of any random numbers generator, because the input to the nonlinear function(s) cannot be easily extracted from the output and vice versa. Unfortunately, there are some drawbacks of the available nonlinear RNGs which can be mentioned as follows [1], [2]:

- The short period of its long cycle length.
- The predefined values of static factors may reduce the associated security.
- Sometimes, the RNG uses one or two nonlinear function(s) during its operation.
- Sometimes, such available of RNGs doesn't satisfied the desire needs for specific applications like change of start and the end of data in the crypto text.

For these reasons, this paper presents a new technique that is called generating random data using 3 nonlinear functions, which are used to generate the data randomly. The technique encourages the extension of the long period cycle length of the repetition and also can be used for specific applications by determining the specific number of digits required to enhance the system security. The rest of the paper is organized as follows. Section (2) presents some remarks on literature review. Main part of the paper (Section 3) is dedicated to several issues related to the proposed technique by combining different operations. This enhances the system security. Section (4) explains the analysis & complexity. Section (5) shows the experimental results. Conclusion & future research assignments will be highlighted in Section (6).

2. LITERATURE REVIEW
This section presents an overview of previous work in RNGs. In general, there are many techniques of RNGs that have been developed by different researchers [1], [5], [6], [7], [14] that may be used in different applications. Some of them used linear functions and the others used nonlinear functions.

To the best of our knowledge, in the exact linear function, it is possible to obtain the output if both the input & operation are known. Moreover, the second input can be obtained if one input & output are known. For instance, the logical operation XOR which acts as a linear function; if one of the inputs is known, then the other inputs can be extracted by performing XOR operation between the known input and output. For instance, if the following output 11010110, and one of the input is 10010100, then the other input is 01000010.

Whereas, in the nonlinear functions, it becomes difficult to obtain the input in a suitable time. The nonlinear functions have been used on the basis of specific mathematical algorithms, which are repeatable and sequential. So, to be useful in simulation, a sequence of random numbers R_1 & R_2 must have two important properties: uniformity and independence. That is, each random number R_i is an independent sample drawn from a continuous uniform distribution between 0 and 1(mean standard deviation 1/2).

Mainly, in the most methods the number m modular divisor should be as large as possible, because a small set of numbers makes the outcome easier to predict. Therefore, the reader can be refereed to some of them which are described as follows [2], [3], [4], [5]:

2.1 Linear Congruential Method (LCM)

This method uses to generate a sequence of integers X_1, X_2,…X_n values between 0 and m-1 by the following recursive relationship as shown in Eq. (1):

$$X_i = (aX_{i-1} + c) \bmod m \quad (1)$$

Such that the four parameters are described as:

X_0 =seed (or starting value)

m =modulus(or divisor)

a =multiplier

c =increment

Where $m > 0$ and $a < m$, $c < m$, $X_0 < m$

The selection of the values for a, c, m, and X_0 drastically affects the statistical properties and the cycle length. The random integers X_i are being generated in the interval [0, m-1]. The main drawback in the LCM is that if an opponent knows the knowledge of a small part of sequence, it is sufficient to determine the parameters of the algorithm.

2.2 Combined Linear Congruential Generators

This method obtains the longer period generator because it combines two or more multiple congruential generators.

- Let $X_{i,1}$, $X_{i,2}$, …, $X_{i,k}$ be the i^{th} output from k different multiplicative congruential generators.
- The j^{th} generator $X_{0,j}$:
- $X_{i+1,j} = (a_j X_i + c_j) \bmod m_j \quad (2)$
- Such that m_j is a prime modulus, a_j is multiplier, and m_j -1 is a period.
- Produces integers $X_{i,j}$ approximate ~ Uniform on $[0, m_j - 1]$.
- $W_{i,j} = X_{i,j-1}$ approximate ~ Uniform on integers on $[0, m_j - 2]$.

2.3 RNG using Cipher Text

This method uses any cipher text to generate random numbers by converting the cipher text to binary digits and selects suitable numbers of binary digits to be converted to decimal digits. This method needs more calculation [2].

2.4 Dynamic Circular Left/Right Shift

The dynamic circular Left Shift is known as the nonlinear function. The main objective of this function is to perform variable circular left/right shift to the mixtures of the data and the secret key. The number of the circular shift/right depends on the position y (i.e. 4 bits) pointed by the value of x, where x in turn depends on the decimal value of the first 'say' five binary bits of the mixture inputs to the function. Because the value of y changes according to the corresponding values of the first five bits of the input, the number of shifts also will change. The value of y ranges from 0 to 15, i.e. the number of shift ranges from 0 to 15. So this function performs variable circular left/right shift which performs variable circular left/right shift operations.

3. PROPOSED TECHNIQUE

This section discus an overall structure of the proposed technique. The main idea of the technique is to combine three nonlinear functions that are used to generate the data randomly by passing an initial value entered by the user as a seed in addition to selected value taken from the buffer. Both of these inputs go through different processes. Firstly, such inputs enter into the XOR function. Then, the output of the XOR operation enters into the combination of three sequential nonlinear functions. One copy of the XOR function output is saved in the buffer which replaces the selected value to be used in next iteration. The output of the combination of the 3 nonlinear functions is divided into the suitable binary digits and converted to data. One copy of the output can be used as a seed to the next round and so on. Fig (1) illustrates the proposed technique block diagram which shows the different operations that have been used. The following subsections describe the 3 nonlinear functions for this technique.

3.1 Dynamic Permutations

The first nonlinear function that is used in this technique is the dynamic permutations, i.e. the permutations are replaced by transpositions which are based on non-predefined positions. This function constructs a suitable hash table along with suitable hash key that divide the binary data into groups. Each group consists of 8 bits, and each 8 bits can take values from 00 to FF in the hexadecimal system. Each group should be hashed into the corresponding value, which is used as an index to store the group in the hash table. Since the values stored in the hash table are based on random indices, each group will take a dynamic position [8]. The output of this function is divided into two parts. The first part can enter into either AND || OR functions. The second part enters into the other function.

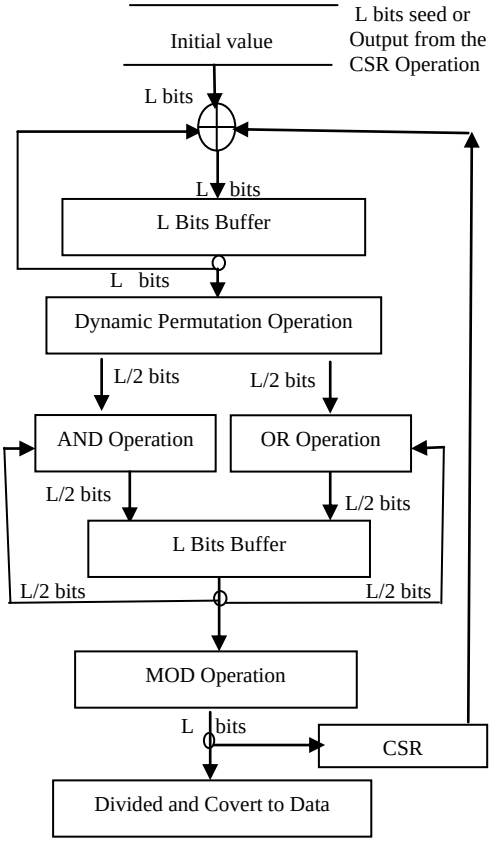

Fig 1: Proposed Technique Block Diagram

3.2 Logical Operations AND & OR

The second nonlinear function that is used in this technique, is the logical operations AND & OR [2]. In the logical operations 'AND' & 'OR' each of them acts as half nonlinear function, that is because nearly half of the input can be extracted from the output. For instance, in the logical AND operation if the following output 10101101 the input is 1x1x11x1, such that x is either 1 or 0. Whereas, in the logical OR operation if the following output 11010110, the input is xx0x0xx0 such that x is either 1 or 0.

Since, the output data L from dynamic permutation is divided into two parts (if one part enters into the AND function, the other part enters into OR function randomly) the size of each one is (L/2). Both functions will take the selected value (L/2) from the buffer. The output from each function is (L/2) concatenated together and then entered into the 'mod' function as L's data input. One copy of the output is saved in a buffer which replaces the selected value in order to be used in the next iteration.

3.3 Mod Operation

The third nonlinear function that is used in this technique is the operation '**mod**' which acts as a nonlinear function. In this function, if one input is known and the output along with the operation '**mod**', the second input cannot be known. For instance, 20 **mod** 6 =2, also 20 **mod** 9 =2, and 20 **mod** 3= 2. The value 2 comes from different operations which are 20 **mod** 6, 20 **mod** 9, and 20 **mod** 3. The output of the 'mod' function is divided into suitable binary digits and is converted to data. One copy of the output recirculates randomly through circulate-shift-right (CSR) and is used as a seed to the next round instead of initial value that is entered by the user as denoted in Fig. (1).

4. ANALYSIS & COMPLEXITY

This section explains the analysis & complexity of the proposed technique. Fig (1) shows the block diagram which denotes the process of generating random data using 3 nonlinear functions. The operations in the required technique consist of logical XOR, dynamic permutations, logical AND & OR operations, substitutions which act as 'mod' operation and circulate-shift-right operation. The processes in these operations are described as follows:

1. Passing L bits initial value (seed); if this value is less than L bits, zeros should be appended to the right most of the block to be L bits.
2. Performing the logical operation XOR between a first L bits initial value entered by the user and L bits predefined initial value from the buffer in the initial iteration, or the value produced from the output of the 3 nonlinear functions operation from the previous iteration. The output of XOR function enter into the dynamic permutations operation. One copy of the output is saved in the buffer to replace the first predefined initial value.
3. Performing the dynamic permutations operation to the output that comes from the XOR operation.
4. Performing the logical operation AND & OR between the output produced by the dynamic permutations and the predefined initial value taken from the buffer. The output will enter into the 'mod' operation. One copy of the output is saved in the buffer to replace the predefined initial value.
5. Performing the mod operation. The output of the mod operation is divided into a suitable number of digits and converted into data. One copy of the output is entered to circulate-shift-right to replace the initial value in the next iteration.
6. Repeating these processes until the required data are generated.

Therefore, as a result, it is clear that the technique consists of logical XOR, dynamic permutations, logical AND & OR operations, substitutions 'mod' operation and circulate-shift-right operation, since each operation needs O(n) time complexity. Then the total time complexity is O(n).

In addition, the selection of the predefined initial value is critical, but it is required only one time during the writing of the computer program.

5. EXPERIMENTAL RESULTS

This section presents the experimental results generated by using our own simulation program for the technique that have been described in the previous sections.

As a result, several tests have been done to perform and examine the technique based on different initial values (text, number, and mix). Generally speaking, the results confirm and generalize that the period cycle length is long. Therefore, there are no repetition of the data item. Hence, the technique increases the security of the system. Tables (1) & (2) illustrates samples of the outputs data for limited digits. Table (1) shows the result of the first initial value which is entered by the user in size of L digits. It presents only the first 10 values of the output. Whereas, table (2) shows the result only for the first 20 values of the output using the

second initial value. The result of first line in table (1) show that the generated digits seem as normally distributed to some extent. For example, in other words, the zero appeared 8 times, the two appeared 8 times and so on.

Table 1. Data Generated by the System for 10 Values.

97417420088418738604 198544752890506180996 469254212624296498 76246662880960970458
118512640592880576684 40332452524212774760 1423 40069252032834263256 260616420912686178696
566888302187482960 458168396448716170240 1169 27039446076246548630 608940332480812364238
102946800802736540354 324600228369407428 972 132940194848868870626 5701321010736564836968
82784866562986184280 104372834518924130856 974 65016053286858424404 2841381019284264898
56674422752670446182 160758850866860398406 2781 9127941702285641016104 824202414352792394384
66880834896518680354 26860222262978584236 2538 79626832299294234562 888128800608654218242
540452974338752924248 29627672699490416882 3244 15836558424262784644 29820280768832960778
2898578706534994840 230990386100050058446 1430 1522002407805287280 94202372888490500810
624934770672854678596 3249829421245686964 1854 920132960988508164646 552264336728260710934

Table 2. Data Generated by the System for 20 Values.

16754958208946992822 4185843886421014948792 1429 390578792300154498550 7687266541768981014990
5501000384946552388668 1269446217032648214 2789 788686496848522240812 284366464632486564528
1898438216882804852 252208994556258712104 2120 6567981787847381002286 824134542336590142244
538482534818532696510 33218416668447640134 3457 146792452932622402672 264224850800710852686
59495830453092238040 8378386972214988224 2343 778384532812396198540 104206308768902798134
610996896514688992640 482720556224768748116 2560 156360278752688548750 8825786108562436062
68682336722720720622 35022686132578512654 297 392250116820956144770 4866097230487638824
78332494464812200216 228888508140204508380 983 091287464816996908 34267899280824830298
55427654222686282692 2429833890096497080 760 134726604681014788996 57437810071216054230
654473435488068466 8416616266258736826 3271 41417046494458630382 63250670112930666328
40682432097026430 198804570354832962602 2096 1506702055236552144 314308532920458340782
626512526434692530526 9861293452412216550 3679 412506436656932610864 256706606320684134856
620482846256512928554 94152394290636140432 3717 398202420784766146994 120700128674766238
82796254160576742626 212958142300246628622 3268 26644449449698494296 8401506287209766401018
50984180896558224708 348476412452574208972 3914 650452810308686436678 79437492488504418550
59047411484879228370 322792461296896852 2342 256772750344308938162 816988356648274534728
82796158466782114998 8132784520510332410 39 89870681244166662574 810366116640550960774
94120434512558122170 661010792232150602506 3474 148526146856446288804 3209009904721036814
6383408528321012908266 4328065566588162514 1892 384924306588308642984 284454624952384942446
2914720994520170394 10071098496644678542 3040 27875893811674612458 360338882776172158250
62690253664596706678 37820422242842342518 283 148958388728162682466 612804692528586118922

6. CONCLUSION AND FUTURE WORK

Generally, a lot of development techniques have been taken place in security systems use nonlinear RNGs to generate secret quantities. Thus, the proposed technique is used to generate the data randomly by using 3 nonlinear functions. The experimental results for different values entered to the system show that, there are no repetition value because the proposed technique encourages the extension of the long period cycle length of the repetition. Therefore, it enhances the security of the system. In future, the technique needs a software application to be compared with other methods.

7. REFERENCES

[1] L'Ecuyer, P. & Simard, R. (2007), "Testu01: A C Library for Empirical Testing of Random Number Generators", ACM Trans. on Mathematical Software 33(4), 22- 27.

[2] Bruce Schneier (2010) “Applied Cryptography” 3rd Ed. John Wiley & Sons. (ASIA) Pvt. Ltd., Singapore.

[3] William Stallings (2009), “Cryptography and Network Security: Principles and Practice” 3rd Ed. India Reprint. Agrawal-M IETE-Technical-Review.

[4] Jerry Banks, etl. (2001), "Discrete-Event System Simulation", 3rd Ed. Pearson Education, Singapore.

[5] Borosh. S. & Niederreiter H., (1983) "Optimal Multipliers For Pseudo-Random Number Generation By The Linear Congruential Method",BIT 23, 65-74.

[6] Figiel, K.D., and Sule. D.R. (Mar. 1985), "New Lagged Product Test for Random Number Generators". Comput. Ind. Eng. Vol. 9, 287-296.

[7] P. L’Ecuyer, “Efficient and portable combined random number generators”, Communications of the ACM 31 June 1988 Volume 31 Number 6, USA.

[8] Saleh N. Abdullah & Sharaf A. Alhomdy (2015) "Dynamic Permutations", Global Journal of Computer Science and Technology (C), Volume 15 Issue 1 Version 1, USA.

[9] Douglasr Stinson (2002) “Cryptography: Theory and Practice”, 2nd Ed. Department of Combinatory and Optimization University of Waterloo, Waterloo.

[10] Behrouz Forouzan (2007), "Data Communications & Networking", 4th Ed., the McGraw-Hill Higher Education, Singapore.

[11] Deborah Russell and G. T. Gangemi Sr (2009) “Computer Security Basics”, O’Reilly& Associates, Inc., New York.

[12] Richard E. Smith. (2011) "Internet Cryptography", Addison-Wesley.

[13] Wright-MA (1999) "Network-Security". Nov., p.11-14 PY: 1999 RT: Journal-article.

[14] Dr. Saleh N. Abdullah & Dr. Sharaf A. Alhomdy, (2015), "Dynamic Random Number Generator based on User Seed(s)" International Journal of Computer Applications, Volume 118 – No. 3, May 2015, New York, USA.

8. AUTHOR PROFILE

Dr. Sharaf Abdulhak Alhomdy, born in 20/01/1971, Alsena, Taiz, Republic of Yemen. Ph.D. in Computer Scince, Pune University, India, 2009. Assistant Prof. & Vice-Dean for students' affare's, Faculty of Computer and Information Technology (FCIT), Sana'a University, Yemen (since 2012). He is an author's of a number of papers. He is promoted to Associate Professor in June 2015, FCIT, Sana'a University.

Dr. Saleh Noman Abdullah Alasaly, born in 1969, Gabel Habashee, Taiz, Republic of Yemen. Ph.D. in Infromation Securty, SRTMU, India, 2005. Asst. Prof., Khawlan College, Sana'a University, Yemen. Head of Information Technology Department, Andlus University, Yemen. He is promoted to Associate Professor in June 2015, FCIT, Sana'a University

Dr. Malek Nasser Algabri, born in 10/4/1981, Alawasga, Sanaa, Republic of Yemen. Ph.D. in Computer Scince, Wuhan University of Tecnology, China, 2013. Asst. Prof. & Head of Computer Science Department, FCIT, Sana'a University, Yemen.