Turkish Journal of Science & Technology
Volume 10(1), 1-8, 2015

The study of the reasons for not giving the results of NIST Tests of Random Walk, Random Walk Variable and Lempel Ziv

[*]E. AVAROĞLU[1], A.B. ÖZER[2], M. TÜRK[3]

[1]Malatya,Turkey
[2]Firat University, Computer Engineering, Faculty of Engineering, 23119 Elazig, Turkey
[3]Firat University, Electric Electronic Engineering, Faculty of Engineering, 23119 Elazig, Turkey
*eavaroglu@gmail.com

(Received: 12.01.2015; Accepted: 23.03.2015)

Abstract

In order to test the statistical proficiencies of the random numbers generated by the random number generators, various test packages such as Crypt-X, Diehard, FIPS 140 and NIST have been developed. Especially NIST package has been mostly used, considered as a valid standard in testing the big random number series. In order for the bit series to be successful in the NIST test, they have to pass all the tests successfully. However, in many studies performed, thirteen results out of sixteen studies were given in many studies performed. The tests of Random Walk, Random Walk Variable and Lempel Ziv Test, whose results were not given, were not announced due to the inadequacy of bit series and the ambiguity. In the study conducted for this reason, these three tests have been explained in detail and a random bit series has been developed by using the linear congruential method. The random bit series generated has been able to pass all the NIST tests successfully.

Keywords: Randomness, Generating Random Numbers, Random Walk, Random Walk Variable, Lempel Ziv

Rasgele Yürüyüş, Rasgele Yürüyüş Değişken ve Lempel Ziv NIST test sonuçlarının verilmeme sebeplerinin incelenmesi

Özet

Rasgele sayı üreteçleri tarafından üretilen rasgele sayıların istatistiki olarak yeterliliklerini test etmek amacıyla Crypt-X, Diehard, FIPS 140 ve NIST gibi çeşitli test paketleri geliştirilmiştir. Özellikle büyük rasgele sayı dizilerinin test edilmesinde geçerli bir standart olarak kabul edilen NIST test paketi çoğunlukla kullanılmaktadır. Üretilen bit dizilerin NIST testinden başarılı olması için tüm testleri başarılı olarak geçmesi gerekir. Ancak yapılan birçok çalışmada on altı testten on üçünün sonucu verilmektedir. Sonucu verilmeyen, Rasgele Yürüyüş, Rasgele Yürüyüş Değişken ve Lempel Ziv Testleri ise yeteri miktarda bit dizisi elde edilememesi ve anlaşılamaması sebebiyle verilmemektedir. Bu amaçla gerçekleştirilen çalışmada bu üç test ayrıntılı olarak açıklanmış olup, doğrusal eşleniksel yöntem kullanılarak rasgele bit dizisi üretilmiştir. Üretilen rasgele bit dizisi tüm NIST testlerinden başarıyla geçmiştir.

Anahtar Kelimeler: Rasgelilik, Rasgele Sayı Üretme, Rasgele Yürüyüş, Rasgele Yürüyüş Değişken, Lempel Ziv

1. Introduction

Random numbers can be defined as numbers which are designated for a certain range, have an equal likelihood to appear and have no certain relation between these numbers. Random numbers have been mostly used in simulations, samplings, numerical analysis, decisions, entertainment, computer programming and cryptography. Random number must provide the following features [1]:

- Unpredictability
- Not being re-generated
- Good Statistical Features
- Uniform Distribution

In order to generate random numbers, different random number generators have been developed such as pseudo random number generators (PRNG), true random number generators (TRNG) and Hybrid random number

generators. Pseudo random number generators generated long random number streams by exposing them to a certain algorithm with any initial (seed) value. True random number generators generate random numbers by using real physical processes as the noise source [2].

In order to test the statistical proficiencies of the random numbers generated by the random number generators, different test packets such as Crypt-X [3], Diehard [4], FIPS 140 [5] and NIST [6] have been developed. Especially in the recent years, NIST packet accepted as a valid standard in testing the big random number streams has been mostly used. NIST test packet consists of 16 tests. In some studies conducted, while the 13 of the 16 tests performed were given, the results of the other 3 tests were not able to be given [7,8]. The reason for this is that the size of the bit streams obtained through the random number generators used is not adequate and the logic of the test were not understood. Thus, in this study, the use of the Random Excursions Test, Random Excursions Variable Test and Lempel Ziv Test will be explained in detail. In order to perform all the NIST tests, 1000000 bit streams was obtained by using Linear Congruential Generator (LCG) [9] and the result of NIST test was given.

In part 2, LCG PRNG is mentioned, which is a way of generating random number and is used to generate random number. In part 3, the explanations of the NIST statistical test were made. In part 4, NIST test results of the number streams obtained are given. In part 5, conclusion is given.

2. Methods of Generating Random Numbers

The generation of random numbers is provided with random number generators. RNGs are systems or devices which are unpredictable, do not repeat itself, have no correlation between each other and produce statistically independent number series. They play an important role in modelling the complex phenomena, sampling, numerical analysis and deciding. Too much storage area must not be used for the numbers generated and too much time must not be needed to reach the desired accuracy. The numbers generated in this way may be random enough for an application whereas they may not be random enough for another application [10].

Random number generators are classified into three categories as pseudo, true and hybrid random number generators.

Pseudo random number generators use a seed value obtained from an entropy [11] source as the random input and generate streams impossible to distinguish from true number generators in terms of calculation. As these generators are deterministic, the output value cannot pass the seed value. Also, the streams start to repeat themselves after a while. That is, the system shows a periodicity. In order to design a system which is not periodic, the memory which the system needs must be quite big. However, this causes the system slow down. PRNGs depend on the safety of the system, the unpredictability of the seed value and the complexity of the functions used in the system. Also, the selection of the parameters used in the functions must be handled with care [2]. Middle-square Method [12], Linear Congruential Generator [12], Data Encryption Standard (DES) [13] and AES [14] are used in the PRNG and some of the known algorithms.

TRNGs are defined as natural physical events which are not deterministic as the entropy source (seed) and used in generating random numbers. They usually include the transformation of analogue source into digital signal. The features and randomness of the numbers generated by the TRNG depend on the randomness of the physical processes. Its disadvantage is that TRNG is slow, costly and depends on the hardware. However, they are used in many applications as they are unpredictable, cannot be re-generated and provide good statistical results [2,15,16].

Hybrid random number generator is the random number generator in which both systems work together using the random number obtained from the TRNG in the PRNG as the seed value.

2.1 Linear Congruential Method

In this study, random bit series is obtained by using LCG which is one of the PRNG methods and whose equation is shown in (1) and (2). LCG is defined as the repeating correlation which will

be used to generate an integer stream such as X_1, X_2, ... between 0 and m-1 [9].

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

$$R_n= X_n/ m \quad n=1,2,..... \quad (2)$$

Random numbers generated in the range of X : (0,m-1)
a : Multiplier
c : increment
m : Mod
R : random numbers obtained between (0,1]

$$S(x) = \begin{cases} 0 & x < 0.5 \\ 1 & x >= 0.5 \end{cases} \quad (3)$$

In the LCG generator used, the seed value was taken as $X_0=102$, a=65536, c=0 and $m=2^{31}$. For the R values obtained, random 1000000 bit streams made up of 0 and 1 was obtained using the rule shown in the Eq. (3).

3. NIST Statistical Tests

By using the valid tests, it was able to be decided whether the bit streams obtained were random. The most valid test known is NIST 800-22, which National Institute of Standards and Technology published. It is made up of 16 tests and explained as follows [6].

- **Frequency (Monobit) Test:** It examines 1 and 0 balance in bit sequence.
- **Frequency Test within a Block:** It is based on observation of number of 1 in m-bit block.
- **Run Test:** It is related with total number of runs in bit sequence.
- Test for the Longest Run of Ones in a Block: It examines lengths of 0 and 1 blocks in sequence. It focuses on the longest group of 1 in m-bit blocks.
- **Binary Matrix Rank Test:** A matrix is formed by using blocks in constant length so that each block defines a line. Linear dependency between blocks is examined by calculating rank of matrix.
- **Discrete Fourier Transform Test:** Discrete Fourier Transformation of the current bit sequence is performed and its periodicity is examined.
- **Non-overlapping Template Matching Test:** It examines whether m-bit block repeats in sequence. If it repeats, new m-bit blocks are formed from the block that is repeated.
- **Overlapping Template Matching Test:** It examines whether m-bit block repeats in the sequence. If it repeats, the block is shifted by 1 bit and new block is formed.
- **Maurer's Universal Statistical Test:** It examines how much the sequence can be compressed without loss of data.
- **Linear Complexity Test:** It examines the complexity of the bit sequence by checking feedback register.
- **Serial Test:** It examines number repetitions of 2^m m-bit repeating blocks.
- **Approximate Entropy Test:** It compares the frequency of two repeating blocks in consecutive lengths (m and m+1) with the expected frequency of a random sequence.
- **Cumulative Sums Test:** This test is based on research of random walk maximal excursions which are defined to be cumulative sum of adjusted (-1, +1) digits in the sequence. The aim of the test is to determine whether cumulative sum of partial subsequences in the tested sequence is too large or too small compared to the expected value of a random sequence. This cumulative sum can be regarded as random walk

The results of the 13 tests explained above are given in many articles. However, the results of the tests of Lempel ziv, Random excursions and Random excursions variants are not given. These 3 tests will be explained in detail in this part [6,17]

3.1. The Test of Lempel Ziv

This test focuses on the number of cumulative separate samples (words) in a stream. The aim is to determine how much a series tested can be compressed. If the stream cannot be significantly compressed, the series is accepted as random [6].

The bit stream generated by ε, RSÜ or PRNG used in calculations ($\varepsilon= \varepsilon_1,\varepsilon_2,.....,\varepsilon_n$), represent

the length of the n bit stream and W_{obs} the number of cumulative different samples (words) which are separated from each other in the series

a) A dictionary is formed according to the rule in Table 2.
ε=010110010

First, as our series consists of 1 and 0, we give 0 and 1 the values of 1 and 2, respectively and the first descriptions of the dictionary are formed.

b) The Lempel Ziv code output sample formed according to the rule in Table 2 is illustrated in Table 1.

Table 1. The Lempel Ziv code output sample

The exit code	1	2	3	4	3	1
The value of dictionary	0	1	01	10	01	0
The sample stream	010110010					

W_{obs}=5 (the number of the values new added to the dictionary)

c) $Pvalue = {}^{1}/_{2}\,erfc(\frac{\mu - W_{obs}}{\sqrt{2\sigma^2}})$ here, μ=69586.25 and $\sigma = \sqrt{70.448718}$ for $n=10^6$. $n \geq 10^6$ must be preferred for the application of Lempel ziv test.

Table 2. The formation of the dictionary

Dictionary	The latest condition	The valid condition	Introduction	The code output	If the value obtained in the introduction part is not in the dictionary, The corresponding values of the values in the last condition are written to the exit code.
0 = 1		ε_0=0			
1 = 2	0	ε_1=1	01=3 (Not in the dictionary - add)	1	
01 = 3	1	ε_2=0	10=4 (Not in the dictionary - add)	2	
10 = 4	0	ε_3=1	01 (Dictionary existing)		
011 = 5	01	ε_4=1	011=5 (Not in the dictionary - add)	3	
100 = 6	1	ε_5=0	10 (Dictionary existing)		
010 = 7	10	ε_6=0	100=6 (Not in the dictionary - add)	4	
	0	ε_7=1	01 (Dictionary existing)		
	01	ε_7=0	010=7 (Not in the dictionary - add)	3	
	0	Finish		1	

3.2. Random Excursions Test

This test focuses on the number of exact K visit cycle in the cumulative sum random walk. Cumulative sum random walk is obtained from the partial sums after organizing the stream made up of (0 and 1s) as the (-1,+1). A random walk cycle starts at a point accepted as random and is made up of a step stream in a certain length until it becomes a complete cycle. The aim of this test is to determine the number of the visits of a certain condition resulting from a deviation expected in the random stream during this cycle. This test is actually a series made up of 8 tests and their inferences [6].

The bit stream generated by ε, RSÜ or PRNG used in calculations (ε= $\varepsilon_1,\varepsilon_2,.....,\varepsilon_n$), represent the length of the n bit series. The reference range used here is the test of chi-square test [6]

a) The stream is normalized. The zeros and ones of input sequence (ε) are turned into -1 and +1 by using $X_i=2\,\varepsilon_i-1$.
For example; ε=0110110101 is X=(-1),1,1,(-1),1,1,(-1),1,(-1),1

b) S_i partial sums are calculated. If Mode is 0, it is started from X_i and if mode is 1then it is started from X_n

$S_1=X_1$
$S_2=X_1+ X_2$
$S_3=X_1+ X_2+ X_3$
.
.
$S_k= X_1+ X_2+ X_3+.....+X_k$
.
.
$S_n=X_1+X_2+X_3+..+X_k+..+X_n$

For the sample in this part,

$S_1= -1$ $S_6= 2$
$S_2= 0$ $S_7= 1$
$S_3= 1$ $S_8= 2$
$S_4= 0$ $S_9= 1$
$S_5= 1$ $S_{10}= 2$

S={-1,0,1,0,1,2,1,2,1,2}

c) By adding 0 to the start and end of the S cluster, S' cluster is obtained. S'=0,s_1,s_2,....s_n,0. For the sample in this part: S'={0,-1,0,1,0,1,2,1,2,1,2,0}. The results of the random walk are illustrated in the Fig. 1.

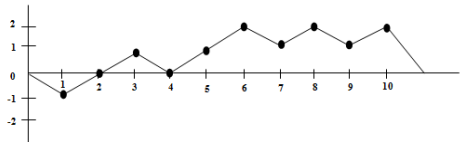

Figure 1. The results of random walk [6].

d) J is the total number of the zero pass in S. Here, zero passes is the number of the zeros seen at first after zero. It forms the cycle between the zero at the start and the next. Then, the end of the first circle will be the start of the second circle and the next cycle is formed as the end of the next zero. If J < 500, the test will not continue. For the example in this part, for S'={0,-1,0,1,0,1,2,1,2,1,2,0} j=3 (the positions of zeros in S are 3,5 and 12). The passes of zero can easily be seen in the graph above. when J=3, there are three cycles here including {0,-1,0}, {0,1,0} and {0,1,2,1,2,1,2,0}

e) As shown in Table 3, the frequency of the x condition values in all circles cycles) is calculated. For $-4\leq x\leq -1$ and $1\leq x\leq 4$. For the sample here, the first circle has 1 -1 formation. In the second circle, it has 1 1 formation and in the third circle, it has 3 1 and 2 formation.

Table 3. The cycles of random walk [6].

Condition x	Cycles		
	Cycle 1 {0,-1,0}	Cycle 2 {0,1,0}	Cycle 3 {0,1,2,1,2,1,2,0}
-4	0	0	0
-3	0	0	0
-2	0	0	0
-1	1	0	0
1	0	1	3
2	0	0	3
3	0	0	0
4	0	0	0

f) For each of the eight condition value of X, for k=0,1,....,5, the total $v_k(x)$ in the cycles of x condition forming k times among all the cycles is calculated.

Pay attention to $\sum_{k=0}^{5} v_k(x) = j$

For the sample here:
The condition of -1 occurs zero time in two circles

- $v_0(-1) = 2$ (The condition of -1 occurs 0 time in 2 cycles) $v_1(-1) = 1$ (The condition of -1 occurs 1 time in 1 cycle) $v_2(-1) = v_3(-1) = v_4(-1) = v_5(-1) = 0$ (The condition of -1 occurs $\{2,3,4,\geq 5\}$ in 0 cycle)
- $v_0(1) = 1$ (The condition of 1 occurs 0 time in 1 cycle)
  $v_1(1) = 1$ (The condition of 1 occurs 1 time in 1 cycle)
  $v_3(1) = 1$ (The condition of 1 occurs 3 times in 1 cycle)
  $v_2(1) = v_4(1) = v_5(1) = 0$ (The condition of 1 occurs $\{2,4,\geq 5\}$ in 0 cycle)
- $v_0(2) = 2$ (The condition of 2 occurs 0 time in 2 cycles)
  $v_3(2) = 1$ (The condition of 2 occurs 3 times in 2 cycles)
  $v_1(2) = v_2(2) = v_4(2) = v_5(2) = 0$ (The condition of 2 occurs $\{1,2,4,\geq 5\}$ in 0 cycle)
- $v_0(-4) = 3$ (The condition of -4 occurs 0 times in 2 cycles)
  $v_1(-4) = v_2(-4) = v_3(-4) = v_4(-4) = v_5(-4) = 0$ (The condition of -1 occurs $\{1,2,3,4,\geq 5\}$ in 0 cycle)

g) For each one of the eight condition of X, $X^2(obs) = \sum_{k=0}^{5} \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)}$ is calculated. Here, $\pi_k(x)$ is the possibility of k times formation of the x condition value in the random range. In the table 4, the possibility of the formation of x condition variable is given.

Table 4. The possibility of the formation of x condition variable [6].

	$\pi_0(x)$	$\pi_1(x)$	$\pi_2(x)$	$\pi_3(x)$	$\pi_4(x)$	$\pi_5(x)$
X=0	0.0	0.0	0.0	0.0	0.0	0.0
X=1	0.50	0.25	0.12	0.06	0.031	0.03
X=2	0.75	0.06	0.04	0.03	0.026	0.07
X=3	0.83	0.02	0.02	0.01	0.016	0.08
X=4	0.87	0.01	0.01	0.01	0.010	0.07
X=5	0.90	0.01	0.009	0.008	0.007	0.06
X=6	0.91	0.006	0.006	0.005	0.005	0.05
X=7	0.92	0.005	0.004	0.004	0.004	0.05

h) For each x value, p value=igamc(5/2, X^2(obs)/2) is calculated. 8 p values are generated.

3.3. Random Excursions Variant Test

In this test, the numbers of visits of the special conditions are measured in the cumulative sum random walk. The aim of the test is to monitor the deviations of the special conditions in the expected visit number. This test is actually the series of 18 conditions (and their results) . The conditions are 9,-8,.....,-1and 1,2,.....9 [6].

The bit stream generated by ε, RSÜ or PRNG used in calculations (ε= $\varepsilon_1,\varepsilon_2,.....,\varepsilon_n$), represent the length of n bit series and the total number of the conditions visited during all the random walks decided on the step d in the test of ξ section 3.2 [6].

a) The series is normalized. By using the (ε) zeros and ones $X_i=2\varepsilon_i-1$ in the introduction series, X_i values are turned as -1 and +1.
For example; ε=0110110101 is X=(-1),1,1,(-1),1,1,(-1),1,(-1),1.

b) Si partial sums are calculated. If Mode is 0, it is started from X_i and if mode is 1then it is started from X_n .

$S_1=X_1$
$S_2=X_1+ X_2$
$S_3=X_1+ X_2+ X_3$
.
.
$S_k= X_1+ X_2+ X_3+.....+X_k$
.
.
$S_n= X_1+ X_2+ X_3+.....+X_k+....+X_n$

For the sample in this part,

S_1= -1 S_6= 2
S_2= 0 S_7= 1
S_3= 1 S_8= 2
S_4= 0 S_9= 1
S_5= 1 S_{10}= 2

S={-1,0,1,0,1,2,1,2,1,2}

c) S' cluster is obtained by adding 0 to the start and end of the S cluster. S'=0,s_1,s_2,....s_n,0.
For the sample in this part: S'={0,-1,0,1,0,1,2,1,2,1,2,0}. The results of the random walk are illustrated in the Fig. 2:

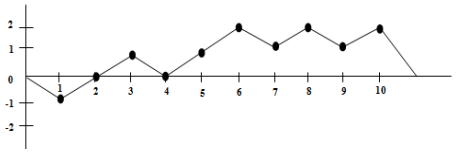

Figure 2. The results of Random excursions variants test [6].

d) For 18 different x condition, the total number ξ(x) of x condition values formed in all j circles is calculated.
For the sample in this part: ξ(-1)=1, ξ(1)=4, ξ(2)=3 and the others are ξ(x)=0.

e) For each ξ(x), $pvalue = erfc(\frac{|\xi(x) - J|}{\sqrt{2J(4|x|-2)}})$ is calculated.

4. The Results of NIST Test

1000000 bit series program obtained by LCG in part 2 was exposed to the test by the program NIST shown with the print screen in the Fig. 3 in [17]. The 16 test results are given in the Table 5.

Table 5. The Test Results of the LCG PRNG NIST

Test Name	P value		Result
Frequency (Monobit)	0.514		Passed
Frequency Test within a Block	0.090		Passed
Runs Test	0.302		Passed
Test for the Longest Run of Ones in a Block	0.349		Passed
Binary Matrix Rank	0.730		Passed
Discrete Fourier Transform	0.818		Passed
Non-overlapping Template Matching	0.645		Passed
Overlapping Template Matching	0.982		Passed
Maurer's Universal Statistical	0.211		Passed
Linear Complexity	0.759		Passed
Serial	0.409		Passed
	0.832		
Lempel Ziv	1		Passed
Approximate Entropy	0.205		Passed
Cumulative Sums	0.886		Passed
Lempel Ziv	1		Passed
Random excursions	-4	0.910	Passed
	-3	0.875	
	-2	0.724	
	-1	0.949	
	1	0.992	
	2	0.976	
	3	0.372	
	4	0.882	
Random excursions variants	-9	0.878	Passed
	-8	0.755	
	-7	0.665	
	-6	0.633	
	-5	0.782	
	-4	0.681	
	-3	0.665	
	-2	0.673	
	-1	0.737	
	1	0.921	
	2	0.882	
	3	0.950	
	4	0.736	
	5	0.566	
	6	0.555	
	7	0.661	
	8	0.790	
	9	0.908	

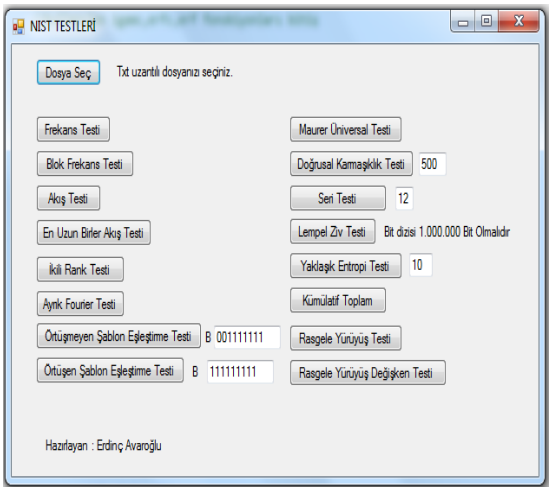

Figure 3. The print screen of the NIST Program [16].

5. Conclusion

In our study, Random Walk, Random Walk Variable and Lempel Ziv Test whose results were not given in the studies in the NIST package conducted to test the statistical proficiencies of the random number series generated by the random number generators have been emphasized. These 3 tests were explained in detail. Also, 1000000 random bit series generation were performed by using LCG PRNG. The random number series were exposed to the NIST test and were found that that their 16 tests results were successful.

6. References

1. Deng and Lin (2000) "Random Number Generation for the New Century", The American Statistician; 54, 2.
2. Koç, C.K. "Cryptographic Engineering", Springer, 2009
3. Caelli, W., Dawson, E., Nielsen, L. and Gustafson, H. (1992) CRYPT–X statistical package manual, measuring the strength of stream and block ciphers
4. Marsaglia, G. (1996) The Marsaglia random number CDROM including the DIEHARD battery of tests of randomness
5. http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm
6. A statistical test suite for random and pseudo random number generators for cryptographic applications; 2010: April [NIST 800-22 Rev 1a]
7. Drutarovsky, M., Simka, M., Fisher V. and Celle, F. (2004) A Simple PLL-Based True Random

Number Generator For Embedded Digital Systems, Computing and Informatics, Vol. 23, 501-515

8. Yıldırım, S. and Bazlamaççı, C.F. (2012) A True Random Number Generator and Test Platform Built in FPGA, 5th Internaitonal Conference on Information Security & Cryptology, Ankara/Turkey, 262-267

9. http://lamar.colostate.edu/~grad511/lcg.pdf

10. Alataş, B., Akın, E. ve Özer A.B. (2007) Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları, XII. Elektrik Elektronik Bilgisayar Biyomedikal Mühendisliği Ulusal Kongresi, Eskişehir Osmangazi Üniversitesi

11. Üstündağ, M., Şengür, A., Gökbulut, G., Ata, F. (2012) Zayıf Radar Sinyallerinde Gürültü Gidermek için Dalgacık Paket Dönüşümü ve Entropi Tabanlı Bir Yöntem, Fırat Üniv. Mühendislik Bilimleri Dergisi 24 (2), 139-147

12. Özkaynak, F. ve Özer A.B. (2006) Lojistik Harita ile Rasgele Sayı Üretilmesi ve İstatistiki Yöntemlerle Sınanması, Journal of İstanbul Kültür University, pp.129-133

13. Data Encryption Standard, FIPS PUB 46-3, Federal Information Processing Standards Publication,1999 http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

14. Kriptoloji Seminer Notları, Şubat 2004, Uygulamalı Matematik Enstitüsü, Kriptografi Bölümü, ODTÜ, Türkiye

15. Kohlbrenner, P. and Gaj, K. (2004) "An embedded true random number generator for FPGAs", FPGA '04 Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays, 71-78

16. Sobotka, J. and Zeman, V. (2011) "Design of the true random numbers generator", Elektrorevue, 2(3):1-6

17. Avaroğlu E., Donanım Tabanlı Rasgele Sayı Üretecinin Gerçekleştirilmesi, Doktora Tezi, Elektrik Elektronik Mühendisliği Fırat Üniversitesi, 2014