ISSN: 0067-2904

Enhancement of Non-Linear Generators and Calculate the Randomness Test for Autocorrelation Property

Ahmed Amer Alsaadi[*1], Ayad G. Naser Al-Shammari[2]
[1]Department of Mathematics, Collage of Science, Directorate General for Vocational Education, Baghdad University, Baghdad, Iraq
[2]Ministry of Education, Baghdad, Iraq

Abstract
In this paper, three main generators are discussed: Linear generator, Geffe generator and Bruer generator. The Geffe and Bruer generators are improved and then calculate the Autocorrelation postulate of randomness test for each generator and compare the obtained result. These properties can be measured deterministically and then compared to statistical expectations using a chi-square test.

Keywords: Cryptography, Stream cipher, LFSR, Key generators, Autocorrelation postulate**.**

تعزيز المولدات غير الخطية وحساب اختبار العشوائية لخاصية الارتباط التلقائي

أحمد عامر السعدي[1] ، إياد ناصر ناصر الشمري[2]
[1]قسم الرياضيات ، كلية العلوم ، جامعة بغداد ، بغداد ، العراق
[2]وزارة التربية والتعليم، المديرية العامة للتعليم المهني ، بغداد ، العراق

الخلاصة
في هذا البحث ، تتم مناقشة ثلاثة مولدات رئيسية: المولد الخطي، مولد جيف ومولد برور. تم تحسين مولدات جيف و برور وحساب خاصية الارتباط الذاتي لاختبار العشوائية لكل المولدات ومقارنة النتائج التي تم الحصول عليها. يمكن قياس هذه الخصائص ومن ثم مقارنتها مع الجداول الإحصائية باستخدام اختبار مربع كاي.

1. Introduction

Cryptographic ciphers classified in two types: symmetric cipher and asymmetric cipher. The symmetric ciphers are moreover divided into stream cipher and block cipher. Block cipher works on big blocks at the same time (usually becomes 128 or 256 bits) and there is no internal state used (at least not used in fundamental version). A stream cipher deals with single bits or single words and needs to keep up an internal state to change the cipher at each move. In general, stream cipher faster than block cipher [1].

The stream cipher is used in many applications in every day (for instance RC4 is used in secure wireless networks and A5 is used in mobile telephones) [1].

In 1976 Professor Solomon W. Golomb had written a book that is an excellent introduction to the theory of shift registers. Ron Rivest, Adi Shamir and Leonard Adleman; in 1987 made a design of a byte-oriented stream cipher called RC4. This kind of cipher found its application in many application security protocols and internet. Moshe Sipper and Marco Tomassini in 1996 published a paper about how to generating a parallel random number by using cellular programming [2]. Cunsheng Ding and Tor Helleseth in 1999 published a paper about describe several classes of the binary sequences with three-level autocorrelation [3].

*Email: Ahmed11math@gmail.com

In 2005 Laszlo Hars and Cortlandt Manor invented a system and method for testing the random numbers generated called the autocorrelation test to determine if the generated random numbers have enough randomness [4]. Yassir Nawaz and Guang Gong in 2008 wrote a paper about how to design a family of stream ciphers which generate a key stream with excellent two-level autocorrelation and the design also contained other randomness properties [5]. Faez H. A. et al., in 2010 published a paper in which that the Golomb's postulates and the binary standard randomness tests are generalized to be suitable to be applied on digital sequences [6]. Benny Y. Zhang and Guang Gong in 2013 published a paper that studied the randomness properties of synchronous stream ciphers that used in wireless communications [7]. Faez et al., in 2017, introduce an attack by employing some weak points in classical stream cipher using Bees algorithm [8].

Ayad G. N. and Amer Abdul Majeed in 2014 published a paper that studied one of basic randomness runs property for stream cipher, it can be determined for any key generator before it is constructed [9]. Abdullah A. G. and Faez H. A. in 2018 published a paper that introduced a new technique in stream cipher based on a dynamic algorithm [10].

2. Stream Cipher Based on Feedback Shift Register

A feedback shift register (FSR) is a consisting of two main parts: shifts register (SR) and its feedback function. The SR is a sequence of bits. Each time only one bit is needed, all the bits in the SR are moved one bit to the right.

The feedback function (polynomial) is simply the XOR mixed the registers.

An FSR is called linear if the connection logic is a linear function on $x_{k-l}, x_{k-l+1}, \dots, x_{k-1}$, that is, if it is of the form

$$x_k = f(x_{k-l}, x_{k-l+1}, \dots, x_{k-1}) = a_l x_{k-l} \oplus a_{l-1} x_{k-l+1} \oplus \dots \oplus a_1 x_{k-1} \quad \dots (1)$$

For some fixed constants $a_1, a_2, \dots, a_l$. Otherwise, it is called non-linear. Here, the values of x_i are either 0 or 1, and hence the sequence is said to be over a binary alphabet, which is usually denoted as f_2, and $a_i \in f_2$ for all i. The operation $\oplus$ can easily be performing as exclusive-OR and $a_i x_{k-i}$ as AND operation both using digital logic gates. In the rest of this paper, we will simply use addition and multiplication (mod2), respectively, for these operations. Over this binary alphabet, therefore, one can add and multiply two elements, and the subtraction is the same as addition. There is just a single non-zero element (which is 1) and the division by 1 is the same as the multiplication by 1 [11, 12].

The simplest form for the FSR is a linear feedback shift register (LFSR) as show in Figure-1.

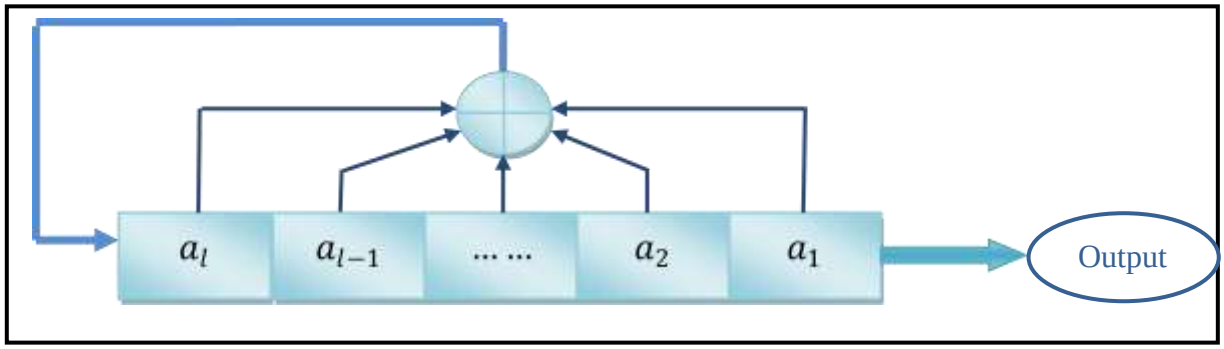

Figure 1- Linear Feedback Shift Register (LFSR).

3. Properties of Pseudo-random Numbers

Before actual algorithms are considered and attempt to evaluate the efficiency of each for cryptographic purposes, it is important that understanding what that are looking for in a cryptographic algorithm [13]. Any good Pseudo-random Numbers Generate will generate a sequence of numbers that have the following properties:

3.1 The Linear complexity

The linear complexity of a finite binary sequence S_j is the length of the shortest LFSR that generates a sequence having S_j as its first terms and denoted by $LC\ (S_j)$ and can by calculated by using Algorithm of Berlekamp-Massey [14].

3.2 The Periodicity

Let the $p(S)$ represents the period of sequence S generate from the LFSR system, and let $p(S_j)$ refers to the period of each sequence generated from LFSR$_j$ for each $1 \leq j \leq N$. Where N is the number of registers, if $p(S_j) = (2^{l_j} - 1)$; where l_j is the lengths of LFSR$_j$; so the periodicity equal to [15]:

$$p(S_j) = l.c.m\left(2^{l_j} - 1\right); j = 1, 2, \dots, N. \quad \dots (2)$$

Where N is the number of combined LFSR's in the cryptosystem.

3.3 Correlation Immunity

It's a relation between the output sequence of Combining Function Unit (CF) $CF = F_k$ from the Key Generate (KG) and the sequences S_k that are combined with each other by CF. This relation occurred because of the high nonlinearity of the F_k. In general the correlation probability (C), which is the ratio between the number of similar binaries (S_k) of two sequences to the length (m) of the part compared of them.

$$CP = S_k/m \quad \dots (3)$$

The correlation immune order can be calculated from the logical truth table for CF depending on computing correlation probability $CP(x)$. For any system the best result of the correlation immunity when $t = k$ (where t is the numbers of immunity LFSR$_j$); that is mean for all x_i, $1 \leq i \leq n$ are statically independent from the output [10].

3.4 Randomness

A random bit generator is a device or an algorithm which it is produced a sequence (of bits) is a statistically independent and unbiased binary digit. In this subsection, Five statistical tests are presented that are commonly used for deciding whether the binary sequence (S) have some specific properties that a truly random sequence. If a sequence passes all five tests, then there is no ensuring that it was indeed produced by a random bit generator. The run, frequency and auto correlation test are called the Main Binary Standard Randomness Tests [16].

Before shedding light on the five basic tests, have to construct the law of Chi-square which is used really.

Assume that the outcome of a random experiment falls into one of *k* categories, and assume by hypothesis that p_i is the probability that the outcome falls into category *i*, assume that *L* independent observation is made, and let Q_i be the number of observation falling into category *i*, in order to test the hypothesis the quantity *T* is compared [17]:

$$T = \sum_{i=1}^{k} \frac{(Q_- - Lp_i)^2}{Lp_i} \quad \dots (4)$$

a)Frequency Test

Frequency test is used to calculate wither the number of 1's and 0's in a sequence S (key stream sequence) with length l are:

$$T_1 = \sum_{j=0}^{1} \frac{(l_j - l/2)^2}{l/2} = \frac{(l_0 - l_1)^2}{l} \quad \dots (5)$$

Where l_1 and l_0 are denoted the number of 1's and 0's in sequence S respectively. The expected value is $l/2$, l is the length of S with freedom degree 1, $T_1 \sim \chi^2(\alpha,1)$.

b)Run Test

A run of sequence S is a subsequence of S consisting of consecutive 1's or 0's, run of 1's called Block while run of 0's called Gap. This run test use to calculate whether the number of runs of various lengths in the sequence S is as expected for a random sequence

$$T_2 = \sum_{j=1}^{m} \frac{(G_j - E_j)^2}{E_j} + \frac{(B_j - E_j)^2}{E_j} \quad \dots (6)$$

Where m equal to the largest block or gap, G_j, B_j be the observed number of gaps and blocks respectively of length j in S for each ; $1 \leq j \leq m$. And The expected value equal to $E_j = (n - j + 3)/2^{j+2}$, n is the length of sequence S, $T_2 \sim \chi^2(0.05, 2m - 2)$.

c)Serial Test

The purpose of this test is to determine whether that the numbers of occurrences of 11, 10, 01 and 00 as a subsequence S are approximately the same as would be expected for random sequence

$$T_3 = \sum_{j=0}^{1} \sum_{i=0}^{1} \frac{(l_{ij} - E)^2}{E} \quad \dots (7)$$

Where l_{00}, l_{10}, l_{01} and l_{11} are refer to the number of 00, 10, 01, 11 respectively and the expected value $= (l-1)/4$, we have $l_{00} + l_{10} + l_{01} + l_{11} = l - 1; T_3 \sim \chi^2(0.05, 3)$.

d)Poker Test

The purpose of the poker test is to determine whether the sequence S of length l each appears approximately the same number of times in S as would be expected for the random sequence. This test work by dividing the S into m parts with length $b \geq 3$ and let l_i be the observed number of occurrence of the i^{th} kind of sequence of length b.

$$T_4 = \sum_{i=0}^{b} \frac{(l_i - E_i)^2}{E_i} \quad \dots (8)$$

Where $E_i = \mathrm{C}_i^b \left(1/2^b\right)(l/b)$, $T_4 \sim \chi^2\ (0.05, b)$.

e)Autocorrelation Test

The purpose of this test is to calculate if there correlations between the sequence S and (non-cyclic) shifted reversions of it. Let τ is a number of shifting S be a fixed integer $1 \leq \tau \leq l/2$.

$$T_5 = \frac{(\, l_0(\tau) - l_1(\tau)\,)^2}{(l-\tau)} \quad \dots (9)$$

Where $l_1(\tau)$ and $l_0(\tau)$ refer to the observed of 1 's and 0 's in shifted S respectively, $T_5 \sim \chi^2\ (0.05{,}1)$.

4. Key Management

To increase the security of the cryptosystem, two kinds of keys are used. These keys are considered as initial LFSRs of the cryptosystem and they are as follows:

1. Message Key (MK): This key is a being send in an encrypted message and will be inserted on some position of the encrypted message upon between the sender and the receiver. It is consists of (8) characters of ASCII CODE (8 bits/character). A new MK can be used with every new message to ensure that no two messages have the same initial key.

2. Basic Key (BK): This key is very important for the security of the cryptosystem and this key must be changed daily. It is consists of (16) characters of ASCII CODE (8 bits/character).

5. Basic Components for cryptosystems

The system consists of the following main components:

i.Basic LFSR's of the cryptosystem

Three basic LFSRs are used:

a)SR1 with length 37 have a characteristic polynomial:

$x^{37} + x^{36} + x^{33} + x^{31} + 1.$

b) SR2 with length 41 have a characteristic polynomial:

$x^{41} + x^{40} + x^{39} + x^{38} + 1.$

c)SR3 with length 53 have a characteristic polynomial:

$x^{53} + x^{52} + x^{51} + x^{47} + 1.$

ii.Enhancement Balance Feedback Shift Register

Balance (to ovoid the correlation) SR with length 61 have a characteristic polynomial:

$x^{61} + x^{60} + x^{59} + x^{56} + 1.$

iii.Feedback functions combining SR

Three combining functions are used, they are as follows:

a)Linear generator that combining function is:

$$fL(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3 \quad \dots (10)$$

b)Geffe generator that combining function is:

$$fG(x_1, x_2, x_3) = x_1 x_2 \oplus x_2 x_3 \oplus x_3 \quad \dots (11)$$

c)The enhancement Geffe generators by adding the balance shift register to the key of Geffe generator and the combining function is:

$$fbG = fG(x_1, x_2, x_3) \oplus x_4 \quad \dots (12)$$

d)Bruer generator that combining function is:

$$fB(x_1, x_2, x_3) = x_1 x_2 \oplus x_2 x_3 \oplus x_1 x_3 \quad \dots (13)$$

e)The enhancement Bruer generators by adding the balance shift register to the key of Bruer generator and the combining function is:

$$fbB = fB(x_1, x_2, x_3) \oplus x_4 \quad \dots (14)$$

The basic components and moving for the proposed cryptosystem are shown in Figure-2.

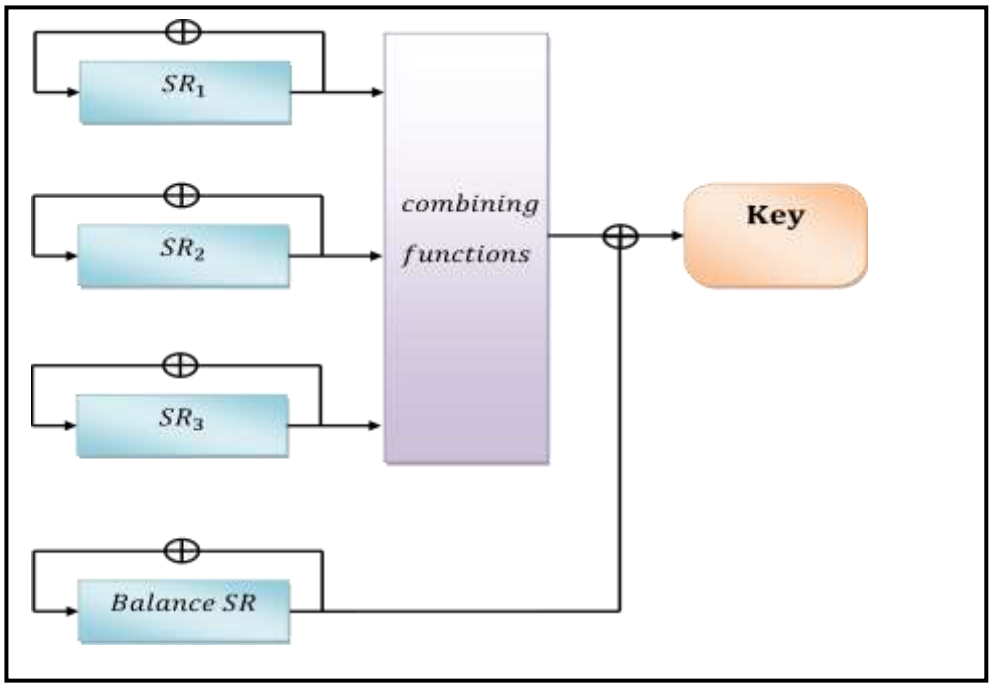

Figure 2- The basic components and moving for the proposed cryptosystem.

6. Initialization of the LFSRs

The main steps of system initialization are:

1- Every character of BK transformed into 8 bits to obtain string of BK bits with total length 128 bits.

$BK_1, BK_2, \dots, BK_i, \dots, BK_{128}$

Where BK_i are the bits of BK, and $1 \leq i \leq 128$.

2- Every character of MK transformed into 8 bits of binary number then the string of MK becomes has total length 64 bits.

$MK_1, MK_2, \dots, MK_j, \dots, MK_{64}$

Where MK_j are the bits of MK, and $1 \leq j \leq 64$.

3. The initial of string shift registers is the result of XOR sum between BK_i and MK_j (without carry) such that:

$BK_1, BK_2, \dots, BK_{64}, BK_{65}, BK_{66}, \dots, BK_{128}$

$MK_1, MK_2, \dots, MK_{64}, MK_1, MK_2, \dots, MK_{64}$

$R_1, R_2, \dots, R_{64}, R_{65}, R_{66}, \dots, R_{128}$

Where $R_i = BK_i \; XOR \; MK_i$ which the string used to fill the (3) LFSR' of the cryptosystem.

4. To complete the initialization process, from R_i string the LFSRs of the cryptosystem are filled one after one as following:

a)SR1 with length 37 bit filled from R_1 to R_{36}.

b)SR2 with length 41 bit filled from R_{37} to R_{76}.

c)SR3 with length 53 bit filled from R_{77} to R_{128}.

For each SR the last stage filled with one (to ovoid zero sequence).

7. Encryption and Decryption Process

For Encryption process, the plaintext characters are converted to L bits (P_i) then xored with output key (K_i) of the proposed cryptosystem to obtain the ciphertext bits (C_i) using equation (15):

$$C_i = P_i \text{ xor } K_i \quad \dots (15)$$

Then the C_i are re-converted to ciphertext characters.

While in the decryption process, the ciphertext characters are converted to L bits (C_i) then xored with output key (K_i) of the proposed cryptosystem to obtain the plaintext bits (P_i) using equation (16):

$$P_i = C_i \text{ xor } K_i \quad \dots (16)$$

Then the P_i are re-converted to plaintext characters.

The main algorithm of the proposed cryptosystem is as follows:

INPUT: basic key (16 characters) and massage key (8 characters), plaintext (ciphertext).

OUTPUT: The output key bits (key sequence) and ciphertext (plaintext).

Step 1: mixing BK and MK to initialize string for shift registers.

Step 2: fill the shift registers one after one.

Step 3: for $j = 1$ to the length of key do steps 4-7

Step 4: for $i = 1$ to 3 do step 5

Step 5: tapping the feedback shift registers to generate a sequence.

Step 6: set key is the combining functions (Linear, Bruer, Geffe) between the three registers.

Step 7: Enhancement of the key generated (Bruer, Geffe) by mixing the key generated with the balance shift register by using XOR combining to obtain (Ki).

Step 8: Ci=Pi xor Ki. (or Pi=Ci xor Ki)

END.

For the output key (Ki) of the proposed cryptosystem, the autocorrelation randomness test can be applied.

7. Test the Results of Proposed Cryptosystem

Example (1): Four samples of keys with different length (L=1000, 5000, 10000 and 50000 bits) are generated from the proposed cryptosystem to calculate the autocorrelation randomness test for these keys and compare the results with distributed chi-square $\chi^2 \sim (\alpha, \nu)$, such that α is the significance level with 0.05 and freedom degree (ν) which is used in the table of chi-square. Tables-(1, 2, 3 and 4) are showing the test results, where P refers to pass the test and F refers to fail in the test.

Table 1-L=1000 bits.

τ	Linear	Decision	Geffe	Decision	Geffe enhanced	Decision	Bruer	Decision	Bruer enhanced	Decision
1	0	P	4.8	F	0.1	P	0.4	P	0.1	P
2	0.2	P	0.5	P	0.2	P	1.3	P	1.2	P
3	3.3	P	0.7	P	3.5	P	0	P	0.3	P
4	0	P	3.6	P	0.6	P	0.2	P	1	P
5	0.2	P	0.2	P	0.5	P	4.8	F	0.8	P
6	0.1	P	0	P	0	P	0.5	P	0.5	P
7	4.8	F	3.5	P	0	P	1.9	P	0.4	P
8	2.3	P	2.3	P	0.3	P	0.2	P	0	P
9	0.2	P	0	P	0.1	P	1.1	P	1.5	P
10	2.5	P	0	P	0.7	P	0.3	P	1.2	P

Table 2- L=5000 bits.

τ	Linear	Decision	Geffe	Decision	Geffe enhanced	Decision	Bruer	Decision	Bruer enhanced	Decision
1	0	P	9.4	F	0.2	P	0.5	P	2	P
2	0.9	P	0.1	P	0.4	P	0	P	1.2	P
3	0.8	P	0	P	0.6	P	0	P	1.4	P
4	0	P	0	P	1.6	P	0.8	P	0	P
5	0.2	P	1.5	P	2	P	0.1	P	0	P
6	0.4	P	0.3	P	0	P	0.4	P	0.3	P
7	1.1	P	1.4	P	0.6	P	1.2	P	0.1	P
8	0.3	P	3.8	P	0.1	P	1.2	P	0.6	P
9	0.3	P	0.4	P	0	P	2.4	P	0.6	P
10	0.9	P	0.5	P	1.7	P	0.5	P	1.8	P

Table 3-L=10000 bits.

τ	Linear	Decision	Geffe	Decision	Geffe enhanced	Decision	Bruer	Decision	Bruer enhanced	Decision
1	0.7	P	0.2	P	1.6	P	0.7	P	2.6	P
2	0	P	0.4	P	0	P	0	P	1.4	P
3	4.2	F	0.2	P	1.8	P	1.6	P	2.1	P
4	0	P	0	P	0.5	P	1.4	P	1.1	P
5	0	P	1.4	P	3.5	P	4.2	F	0.8	P
6	2.9	P	0.1	P	0	P	0.2	P	0.5	P
7	0	P	2	P	0	P	5.1	F	0	P
8	0.1	P	2.4	P	0.3	P	2.6	P	0.2	P
9	0	P	1.6	P	0	P	0.9	P	2.1	P
10	0.9	P	1.6	P	3.7	P	0	P	1.4	P

Table 4- L=50000 bits.

τ	Linear	Decision	Geffe	Decision	Geffe enhanced	Decision	Bruer	Decision	Bruer enhanced	Decision
1	1.6	P	0	P	2.1	P	0.2	P	2.2	P
2	0.1	P	0	P	2.4	P	2.5	P	0.7	P
3	0	P	0.4	P	0.6	P	0.3	P	0.1	P
4	3.2	P	0.4	P	0	P	1.7	P	0.6	P
5	0.9	P	0	P	0	P	0.1	P	0.1	P
6	0	P	1.8	P	1.4	P	0.7	P	0.1	P
7	2.9	P	1	P	1.4	P	0	P	0.2	P
8	1.9	P	0	P	0	P	2	P	1.9	P
9	0.7	P	0.8	P	0	P	0	P	1.1	P
10	3	P	0.6	P	2.8	P	1.9	P	2.3	P

8. Conclusions

Notice that from the above example some important results that are summarize:

1.The autocorrelation is very important test, since it can be transformed to frequency when $\tau=0$, and to Poker test when $\tau=0$ and $k=1$ (k is the length of tested string of S).

2.The key length effect on the result of the test, when increasing length key the result will be improved to better.
3.The enhancement key generators Geffe and Bruer give the best result from the unimproved generators, such that the enhancement key does not give any fail in the test.
4.The characters of both basic key and message key must be different characters (not the same used in both keys).
5.The enhancement cryptosystem can be used to encrypt texts, images, videos, and any media files, since it's can make a long length high, security and speed.
6.As future work, since the proposed cryptosystem consists from (3) subcryptosystems, so it can be dynamic in each run by using different cryptosystem each time.
7.We recommended using the other statistical randomness tests for the cryptosystem.

References

1. Andreas Klein. **2013.** "*Stream Ciphers*", Springer-Verlag London.
2. Moshe Sipper. **1996.** "*Generating Parallel Random Number Generators by Cellular Programming*", World Scientific Publishing Company.
3. Cunsheng Ding. **1999.** "*Several Classes of Binary Sequences with Three-Level Autocorrelation*", IEEE Transactions on information theory.
4. Laszlo Hars and Cortlandt Manor. **2005.** "*Randomness Test Utilizing Auto-correlation*", United States Patent.
5. Yassir Nawaz and Guang Gong. **2008.** "*A family of stream ciphers with designed randomness properties*", Elsevier Inc.
6. Faez H. A., Sahar A. M. and Mahmood A. Shamran. **2010.** "*Generalize the Randomness Tests to Test the Digital Sequences Produced from Digital Stream Cipher Systems*", Iraqi Journal of Science.
7. Benny Y. Zhang. **2013.** "*Randomness Properties of Stream Ciphers for Wireless Communications*", IEEE, University of Waterloo, Canada.
8. Faez Hassan Ali, Mohammed G. S. Al-Safi and Ahmed Ayyoub Yousif, **2017,** *"Analyzing Cryptosystems by Using Artificial Intelligence"*, Special Issus: 1st Scientific International Conference, College of Science, Al-Nahrain University, 21-22/11/2017, Part I, pp.100-108, DOI. 10.22401/SIC.I.14.
9. Ayad G. Al-Shimmari and Amer Abdulmajeed**, 2014.** *"*A comparative study to calculate the Runs property in the encryption system*"*, *Journal of College of Education for Women*, **25**(4).
10. Abdullah A. G. and Faez H. A. **2018.** "Robust and Efficient Dynamic Stream Cipher Cryptosystem", *Iraqi Journal of Science*, **59**(2C): 1105-1114, DOI:10.24996/ijs.2018.59.2C.15.
11. Bart Preneel. **2008.** *" Cryptanalysis and Design of Stream Ciphers",* Katholieke University Leuven, Heverlee (Belgium).
12. Rob Curley. **2013.** "*Cryptography Cracking Codes*", Britannica Educational Publishing and Rosen Educational Services, LLC.
13. Hans Delfs and Helmut Knebl. **2015.** "*Introduction to Cryptography Principles and Applications",* Springer-Verlag Berlin Heidelberg.
14. John Talbot and Dominic welsh. **2006.** "*Complexity and Cryptography*", Cambridge University Press, New York.
15. Hu Xiong, Zhen Qin and Athanasios V. Vasilakos. **2017.** "*Introduction to Certificate less Cryptography*", Taylor & Francis Group, LLC.
16. Andrew Rukhin, Juan Soto and James Nechvatal. **2010.** "*A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*", National Institute of Standards and Technology.
17. Ayad G. N. Al-Shimmari, **2009.** "*Mathematical Modeling and Analysis Technique of Stream Cipher Cryptosystems*", Ph. D. Thesis, Department of Applied Science and Department of Computer Science, the University of Technology.