The Robust Stream Cipher for Securing Data in the Smartphones

Abdullah S. Abid

Department of Computer Technique Engineering, Engineering Technical College, Middle Technical University

orjent_2007@yahoo.com

Mohammed J. Zaiter

Department of Computer Technique Engineering, Engineering Technical College, Middle Technical University

mjzaiter@eetc.mtu.edu.iq

Tayseer S. Atia

Department of Computer Engineering, College of Engineering, Al-Iraqia University

tayseer_salman@yahoo.com

Submission date:- 26/12/2018	Acceptance date:- 30/1/2019	Publication date:-3/2/2019

Abstract

With the development of network and communication systems in large areas in the world, this leads to increase security problems in transmission of data such as data leakage, modification, unauthorized access, and attacks. There are many types of techniques that are used to prevent these problems and protect data. One of these techniques is a stream cipher which considered the strongest and fastest method used in encryption and decryption process. In this study presented a new design for the stream cipher to protect mobile data. The strength of stream cipher depends on it is' key. There are several methods to generate key. We used three types of generator. Then, it used the combiner to convert them into a nonlinear Boolean function in order to make the generator key more secure. To implement a new generator key by using these three kinds, we used four LFSRs and one of NLFSRs or FCSRs to produce five variables Boolean function. These variables will be as an input to the combiner function. Finally, we tested the generator and submitted it to the randomness tests that is publicly available in the National Institute of Standards and Technology (NIST).

key word: Stream cipher, LFSR, FCSR, NLFSR, Non-Linear combination Boolean Function, NIST tests

1. Introduction:

In recent years, communication networking is spread in wide areas of the lands, where data transmission between devices became more susceptible to infection for many attacks. It should protect the data, where many researches are interested to studying the field of data security. Obviously, to design an efficient system for protect the data through encryption them, so it needed to use a kind of cryptography with fast in processing and as little cost as possible. Where consider the best method for encryption of data is stream cipher because it is very fast to implementation in software and hardware as well as it is more suitable for many applications and devices [1]. There are some of papers works in lightweight stream cipher like [2], [3] and [4].

Stream ciphers are a category of symmetric cryptography. It means that, it is using a same key for a process of sending (encryption) and receiving (decryption). Where considered the pseudo random number generator is the most common way to generate key (or keystream), after that it is using XOR with the plaintext to produce ciphertext [1]. Several algorithms of stream ciphers are proposed in [5]-[8], some algorithms have proved effective against attacks. In this paper, it proposed the new design of the stream ciphers that suitable for mobile communication systems. Hence, we are using some types of pseudo random number generator are LFSRs, FCSRs, and NLFSRs, after that it combined them into a combination functions to generate a keystream. Also, we used a new idea is a comparator. It put in between the NLFSRs and FCSRs to make the generator more robust due these two types are new and it difficult to analysis them so far.

The remainder of the paper is arranged as follows: Section 2 described the types of generator that it used to generate a keystream by explaining the principle work it in briefly. Section 3 described the proposed keystream generator design and we mentioned the properties for each type of registers as well as the combination functions. Section 4 the execution of the proposed of keystream was evaluated by using NIST. Section 5, we mentioned the conclusion from this paper.

2. Background Theoretical:

2.1 Linear Feedback Shift Registers (LFSRs):

LFSRs are most widely recognized kind of shift registers that utilized as a part of cryptography. LFSRs consist of two parts as shown in Figure 1. One of them calls a shift register and the other calls feedback function. Shift register is a set of cells or stages that store one bit (1 or 0). The initial states of the LFSR called a seed. These bits are shifted to the right simultaneously by the external clock that is responsible for control of the movement of these bits. In addition, the new bit in the left most of that register is computed by the feedback function. The feedback function basically formed by the exclusive-or (XOR) of specific bits in the register, where these bits called a tap sequence (or Fibonacci configuration) [7]. When implementing any register from type of LFSR, so it will be represented by using connection polynomial equation as shown in below [8].

$$P(x) = 1+\sum_{i=1}^{L} ciX^{i}. \qquad (1)$$

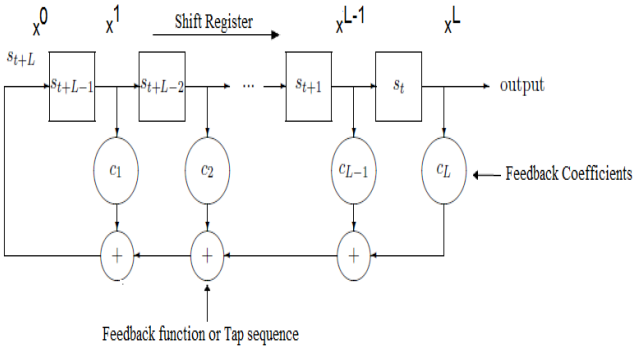

Figure 1: Linear Feedback Shift Registers [8]

LFSRs represent a basic component to generate a sequence of bits as a keystream, where it considers very suitably for the stream cipher applications. The implementation of the hardware as well as the software is very easy, and they require less cost and time to be implemented. Also, it produces a sequence of bits with excellent statistical features.

2.2 Feedback with Carry Shift Registers (FCSRs):

FCSRs are like LFSRs. Both of them contain two parts which are a shift register and feedback function. However, FCSRs have an additional part which is called a carry register as shown in Figure 2. Moreover, FCSRs do not use (XOR) in their function, but they used a function that called summation of integers. Where the summation of integers is formed by adding the values of the taps of sequence (active taps) with the value of carry register. The result of previous process is an integer and calls also a parity bit (σ). Then, the new bit or state can be calculated by taking mod two to the parity bit (σ mod2). Moreover, new carry bit calculated by using the equation. (⌊σ /2⌋), where the ⌊ ⌋ is the greatest integer or integer part [7].

When constructing any FCSR with excelent properties, then using the next equations. FCSR will represent by using an equation that it called a connection integer (q). it is' define in section 3 see definition 3.1 in [16] , where r is a taps q1, q2, q3, . . . , qr and it describes the number of active taps by the equation below [16].

$$\mathbf{q}=q_r 2^r + q_{r-1}2^{r-1} + \cdots + q_1 2 - 1 \qquad \textbf{(2)}$$

Where q should be a prime number

Also, it can determine the number of cells (stages) for the FCSRs by using the next equation [16].

$$\lfloor \log_2 q + 1 \rfloor \qquad (3)$$

Morever, through the result of the eq no.3, it can calculate the number of carry by the equation [16].

$$\lfloor \log_2 r \rfloor \qquad (4)$$

Finally, it must be chosen the initial states be carefully for this register. If the output of sequences s = $(s_0, s_1, s_2, ...)$ that generated by the FCSRs with q is periodic, and if y = 2^{-1} is the multiplicative inverse of q , then we conclude the output of sequence by the equation [16].

$$a_n = \left(y^n (mod\ q)\right)(mod 2) \qquad \textbf{(5)}$$

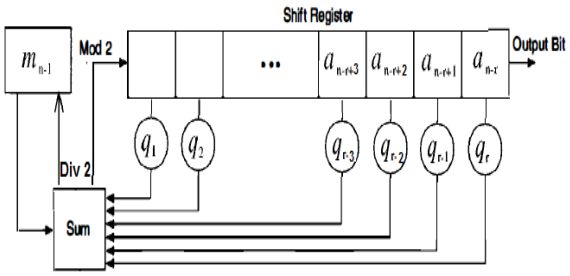

Figure 2: Feedback with Carry Shift Registers [15]

FCSRs ,as memtioned in [7], is a new generator that can be used to produce a random key. It used a new technique in the arithmetic called a 2-adic number instead of using the arithmetic in **F**2 in both type of LFSRs and NLFSRs (where the **F**2 is a finite field with the binary numbers (1 and 0)). It gives a good statistical property and can be resist for many attacks to provide a built-in in nonlinearity.

2.3 Nonlinear Feedback Shift Register (NLFSR):

NLFSR is another type of generator that is used to construct a keystream as a form of pseudo random generator as shown in Figure 3. In recent years, the NLFSRs have gotten much consideration by researchers to design various novel cryptographic algorithms. The reason for that is that NLFSRs are more secure and stronger than LFSRs to be breakdown by existing cryptanalysis techniques. The state of

NLFSRs contain nonlinear function that provide that security properties. However, NLFSR has many drawbacks. One of them is the output of the sequence may not always equal to the length of the maximal period expected. The other drawback is the output of bits for the type (n,k)-NLFSR with the period $2^L - 1$ does not achieve the first and second hypotheses of Golomb. These problems can be addressed by using the Fibonacci configuration [9] [10].

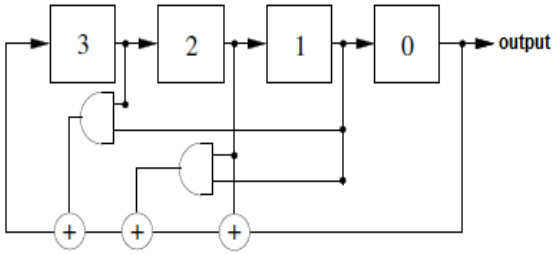

Figure 3: An example of Fibonacci NLFSRs [9].

Actually, NLFSRs consider the best methods and most complex to design keystream than the other types (FCSRs and LFSRs). In fact, it is not found any method or mathematical theory can analysis them until now. So, it is considered one of the most powerful ways to design the keystream [7]. In this paper, it can be used one of types NLFSRs that is called a Fibonacci NLFSRs instead of used a Galois NLFSRs because it contains two main advantages. Firstly, any output of sequence is always achieve the first and second of Golomb's postulates [11] with the cycle of $2^L - 1$, while the Galois (L,k)-NLFSRs does not achieve any of first and second of Golomb's postulates. Secondly, the output of sequence always equal the length of maximal period, while the Galois (L,k)-NLFSRs is not necessarily equal the length of maximum period [12], [10]. In other word,NLFSR Fibonacci give a pure cycles if the feedback function is a type of equation as shown below.

$$\boldsymbol{f(s_0, s_1, \dots, s_{n-1}) = x_0 \oplus g(s_0, s_1, \dots, s_{n-1})} \qquad \textbf{(6)}$$

Where g is not depended on the value s_0. Due f is a type of DeBruijn, So the states $(s_0 s_1 \dots s_{n-1})$ and $(\overline{s}_0 s_1 \dots s_{n-1})$ each of them, the inputs have a different value because all the cases are depended on the value g. In the other meaning the value of g is the same in all cases but the values of s_0 and $\overline{s}_0$ are different so that f cannot be a type of above equation no.6.

3. The proposed model

This section depicted in details the structure of the new design for the stream cipher to generate key on a form of Pseudo-Random Number Generator (PRNG).The security in the stream cipher depends on it is' key. It must be random to become more complex to be breakdown and hard to be predicated by attackers.

The proposed structure appears in Figure 4. It comprises of two parts. The first part is inputs and the second part called a Combination Boolean Function with five variables.

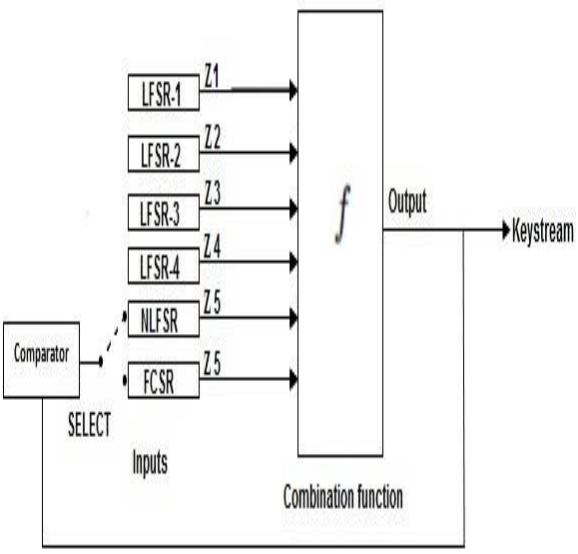

Figure 4: Proposed design for key generator

3.1 Input Part:

The first part is inputs that consist of many types of generators which are LFSRs, NLFSRs, and FCSRs. Inputs part contains of three types of generators to create a keystream. In this part we will explain properties, and specifications for each kind as shown below:

3.1.1 Properties of LFSR Sequences:

LFSR has many characteristics to generate a good sequence with perfect statistical properties. All the characteristics below were used to implement our 4-LFSRs generators.

1-LFSRs by the feedback coefficient produce a sequence of different length, where was represented by the polynomials (connection polynomial) as shown in the equation (1)

2- LFSRs generate sequence of bits with maximal period (longest pseudo random sequence before it repeated) to give us a sequence with good statistical properties. It must be used a type of equation called primitive polynomials, so the maximum sequence equal $2^L - 1$ [5]-[8]. It preferably use LFSRs from a type of non-singular to be periodic. In other words, the degree of non-singular for feedback polynomial is equal to the LFSRs length. it will ignore any type of singular LFSRs because the generator will become ultimately periodic. [8] [5]

3- At the design, it should use a primitive polynomial that is a type of dense (not sparse) to improve a process of security for communication systems and applications in order to make attacks more difficult to penetrate it [8] [5] [13]

4- Linear complexity (L.C): linear span is the length of shortest LFSRs that it generates the output of sequence (S). The linear complexity should be large to make detection the length of registers extremely difficult and better at the design of our-generator. So, L.C was constructed very high.

Although the LFSRs are fast in implementation hardware and software and give a long period and good statistical properties but it is very weak to be hacked because it is very linear and it can detece the number of taps by using Berlekamp-Massey algorithm [14] [15].

3.1.2 Properties of FCSR sequences

In the following points, some characteristics were displayed which were used to implement this generator in our design.These points include knowing how to calculate the number of stages & carry for this register. In addition to, how to set the initial states for purpose of generating the random sequences.

1- FCSRs was represented by the connection integer (q) as shown in eq no.2.Through the connection integer (q), the number of stages in the FCSRs were determined by the equation no (3). In addition, the number of carry was determined (by using eq no.4) via the number of cells through the result of eq no.3. The type of q was used in our design to be a prime number in order to generate a maximum period.

2- From the eq no.5, we can benefit to determine the initial loading of FCSRs, so when we generate the sequence for best long period, the place of the initial value for the register + carry of the FCSRs must be approached with carefully.

3- To production a maximum possible sequence of FCSRs, so the period equals q-1. Or it can be calculated the period of FCSRs by $\varphi(q)$ (it is the Euler's phi function and it equals the numbers of integers which is less than q and the GCD with q = 1). See the definition13.1 in [16].

4-Any formal power series = $\sum_{i=0}^{\infty} s^i 2^i$, which is 2-adic numbers can be giving us an infinite binary sequence s = ($s_0, s_1, s_2, \ldots s_\infty$). The binary sequence (s) it will be a periodic if and only the 2-adic number is the rational number ($\alpha = \frac{r}{q}$). Where r and q are integers, $\alpha < 0$,and $|r|$ it must be < $|q|$.

5- The 2-adic span is the smallest size of FCSRs that generates the output of sequence (s). The range of 2-adic identify by an effective method by using the 2-adic approximation theory (see [16] in section 10). There is an algorithm to find the smallest FCSRs that generate the sequence a. It works with knowledge just of 2y+2log(y) the sequence of bits s (where y is the 2-adci span of s). This algorithm depended upon the de-Weger's theory [17].

3.1.3 Properties of NLFSR sequences.

Fibonacci-NLFSRs contain many important properties for producing a sequence with a huge period. It illustrates as in the following points [9].

1-NLFSR Fibonacci has pure cycles, so it was used in our design to be as a form of eq no.6 (see section 2.3).

2- Fibonacci NLFSRs have $2^{2^{L-1}-L+1}$ different L-bit with the sequence equal $2^L - 1$. This formula can be derived as follows, where *Gn* has 2^L nodes, representing all possible states of an L-bit NLFSR, and 2^{L+1} edges that representing all possible transition between these states. Each node of *Gn* has two possible outputs and two possible inputs.

3- The form of cycle of length $2^L - 1$ is come from a cycle of length 2^n; we can delete the loop at node 00...0 and also the loop at node 11...1 of Gn. Since there are no other loops in Gn, there are precisely two cycles of length $2^L - 1$for each cycle of length 2^L. Therefore, the number of cycles of the length $2^L - 1$ in the Gn equal $2.2^{2^{L-1}-L}$.

3.2 Second Part (Combination Functions (f))

The previous types like LFSRs and FCSRs, when we generate keystream, it contains on a drawback. It is a linear, so it is very easy to detect the length of each register and the number of taps by using multi algorithms attacks such as Berlekamp-Massey algorithm. To avoid these problems, it should be using the types of nonlinear generator technique, either nonlinear combination generator, or nonlinear filter generator, or clock-controlled generator. In this paper, we used a combination generator to destroy the linearity in the different types of generator key. A combination generator is used in multiple stream cipher applications. It consists of several type of running-key generator in parallel that combine them to produce the keystream, as in the Figure 4 above.

A product of any result Boolean function (f) called an m^{th}(resilient Boolean function). The expression of f (Z1, Z2, . . . , Zn) is represented by the algebraic normal form (ANF) of f [18] .It contain of four parameters are number of variables, resiliency, algebraic degree, and nonlinearity (n,m,d,x). A resilient

Boolean function in stream cipher must satisfy several criteria in order to resistant many attacks as possible in the same time and generate good stream cipher. These criteria are balance, high nonlinearity, high algebraic degree, and high algebraic immunity [19] [20]. There are several interested researches to construct Resilient Boolean Functions with best cryptographic properties [19]-[22]

Properties of Resilient Boolean Function

1- Balancedness

Boolean function f is balance, if Wt(f) =$2^L - 1$, where Wt(f) is the Hamming weight of a Boolean function f(z) of N-variables [19], [21], [22]. In other word, the Boolean function f is balance if the number of one's equal the number of zeros {z|f(z)=0}={ z|f(z)=1}.

2-Nonlinearity

Boolean Function f focus to produce a capacity accomplishing as high nonlinearity, if the Boolean function is balance, then the n-variables for odd (n) having a nonlinearity equal

$$\mathbf{2^L - 2^{L-1/2}} \qquad \mathbf{(7)}$$

If the Boolean function is balance, then the nonlinear for functions $\boldsymbol{f}$ is at most n-m-1, for $1 \leq m \leq n-2$ [5].

3- Algebraic Degree.

Algebraic degree of the Boolean function f must be high, since all cryptosystems utilizing Boolean functions f can be assaulted if the functions have low algebraic degrees. When the algebraic degree is increased, the linear complexity will also increase, so the design of generator will be more powerful and hence the Berlekamp-Massey calculation turns out to be computationally infeasible. The algebraic degree is not exceeding of $d \leq n-m-1$ [8], [19].

4- Correlation Immunity.

The correlation immunity of f must be high. The meaning of correlation immunity is ensures the output of the Boolean function f cannot leakage information about the variables for each inputs. There are tradeoffs between high correlation immunity, high algebraic degree, and high nonlinearity.

There are some attacks infect the combination functions such as correlation attacks, fast correlation attacks, and algebraic attacks. The principle work of these attacks are tries detect the content of each register. If the length of each register and the combination functions are known, then the secret thing is the initial states. Therefore, most of the attacks on the combination functions works to identify the initial states of all types of registers through exploits presence the statistical dependence between the output of single register and the keystream. To make these attacks are practically useless, it must be used the four-LFSRs in the design from a type of not sparse. Also, it must be increasing the Correlation Immunity (CI) in the combining function when the function is balanced. Moreover the nonlinearity must be high to prevent the correlation attacks and fast correlation attacks from penetration [8] [5].

In the Figure 4 (proposed structure for generator key) of this paper, we used the inputs (registers) within the specifications required to generate the key and which mentioned in section 3.1.1,3.1.2 and 3.1.3.In the design it should be the GCD between each register equal to 1 [8]. Also the NLFSRs and FCSRs, we select the same length. The length of NLFSR we selected it from [9], and FCSR we selected his length from [7]. The combination function with five variables (5,1,3,12) selected from [23]. It must be select five from six variables through the comparator, where we place it in the NLFSR and FCSR to select one of them every time of period, For example every 100 period. So we divided the inputs into two group, group A include the four LFSRs with NLFSR, and group B include the four LFSRs with FCSR. The principle work of comparator depended on the contents of each two registers. If the output of combination function equal 1, it will comparator of the greatest value or equal in the contents of each register and vice versa it will compare the smallest value. The utility of using comparator is working to prevent of detect the initial states for NLFSR and FCSRs and hence the attacker can not to determine any group is switch on.

4. Results:

This section tested the generator key and displayed the experimental results by using National Institute of Standards and Technology (NIST) that explained in [24]. The NIST test suite is statistical bundles consist of few tests. The purpose of NIST test is to test the randomness of binary sequences that is produce through pseudorandom number generator.

4.1. Result of the NIST Statistical Test:

In this section, we will display the result of NIST, where we tested five tests of NIST on the proposed keystream generator design (see Figure 4 in section 3). The sequence n has successfully passed in the five tests of NIST tests, and we calculated the P-value for each test and the result is > 0.01. It concludes from this design, it has excellent statistical properties and it considers random. Also, it can resist many known attacks.

4.1.1. Result of the Frequency (Monobit) Test

Figure 5 demonstrates the relationship between p-value (in y-axis) and number of bits (in x-axis). We set the value of number of bits (n) to be between 100 to 10^6bits. P-value was almost 1 when n=100. However, p-value was 0.15 when the n =500. In both cases, p-values were larger than 0.1, and this is what we would like to see. Because as mentioned in [24], if the p-value is larger than 0.01, then the key is randomness enough and it is very hard to be broke. When we set number of bits to be 103 and 105 consecutively, the p-value were close to 0.9 which is very good result. Moreover, when we chose the number of bits to be 6*105, the p-value was close to 0.7. In addition, when we set number of bits to be 2*105, 3*105, 4*105, 7*105, 8*105, 9*105, and 106, the results of p-value were fall between 0.4 and 0.6. Finally, all of points passed this test because all p-values' points were more than 0.01.

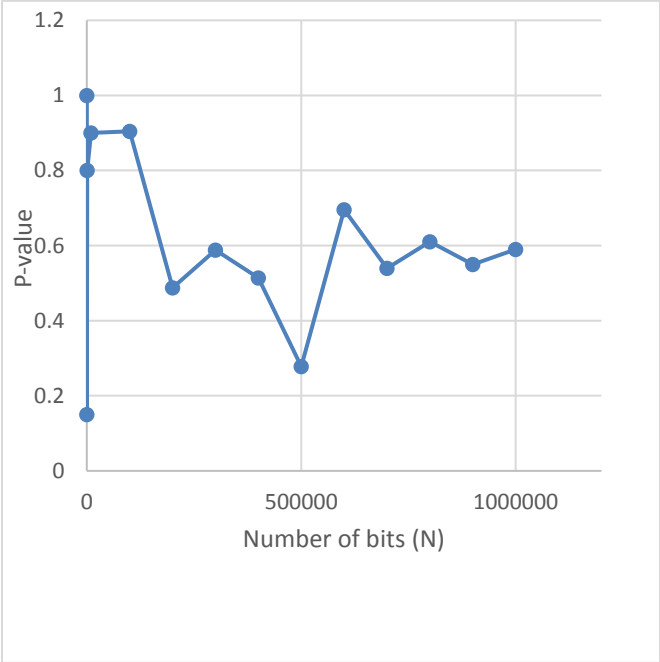

Figure 5:The p-value for The Frequency (Monobit) Test

4.1.2 Result for Frequency Test within a Block:

The figure 6 in below illustrates the results of the frequency test within a block. It notices that the output of sequence for n=100, the p-value= 0.616. While the other sequences, the P-value is stable at the value 1.

Note: When we using this test for many output of sequence, the value M (number of bits for each block) is change and it should be $M \geq 0.1n$ and the number of blocks $N<100$ [24]. For instance, when we test the sequence n =100 the value of number of bits in each block is equal M = (0.1*100)=10, and as in the following sequences.

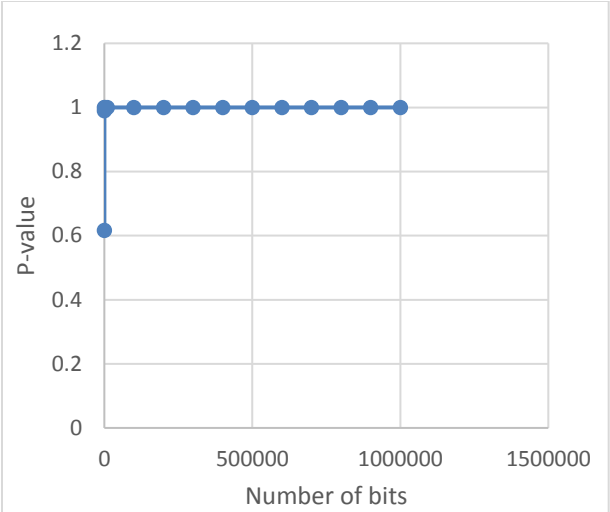

Figure 6: The P-value for The Frequency Test within a Block

4.1.3. Result for Runs Test

In this test, we used several values for the number of bits that is recommended for the NIST. We started ascending at $n \geq 100$. In this chart, we notice that all values used have successfully passed the test and are considered random. For example when we generate a sequence of bits (n) equal 100 bits so we notice that the p-value = 0.07. And also when doubling n into 200 bits the p-value has increased to 0.14. Also, when we took n = 500 bits the p-value increased more to become 0.85. In this Figure (7), we notice that the peak value of p equal approximately to 1 when we used n equal 1000 and 10000 bits. we tested the other bits (n) that equal between 10^5 to 10^6 bits, we found that the most p-value ranging from 0.07 to 0.41.

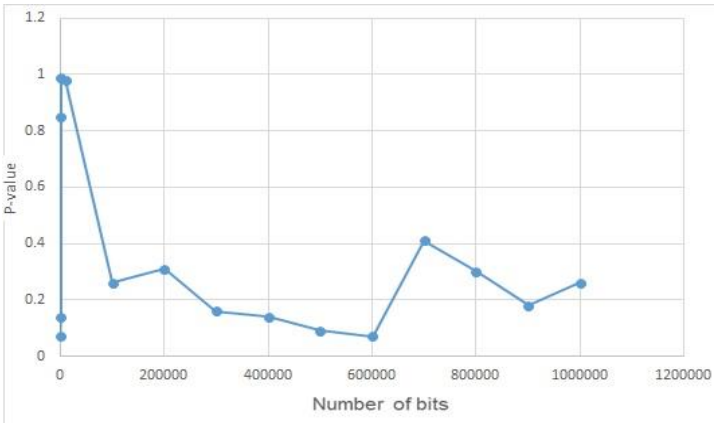

Figure.7: The P-value for the Runs Test

4.1.4. Result for the Longest Run of Ones in a Block

In this test, we show the results by two figures. Where we take three lengths of n are 128, 6272, and 750000 at minimum and the M equal 8, 128, and10^4. The result of the p-value in the first n =128 is equal to 0.23. The results for the length of the others are illustrate in the two figures in below. Where the M is constant (128, and10^4) and we changed the lengths from 6272 to 10^6 in Figure 8 and 750000 to 10^6 in Figure 9. All the values that selected in the two figures were the p-value greater than 0.01, so all the sequences are random.

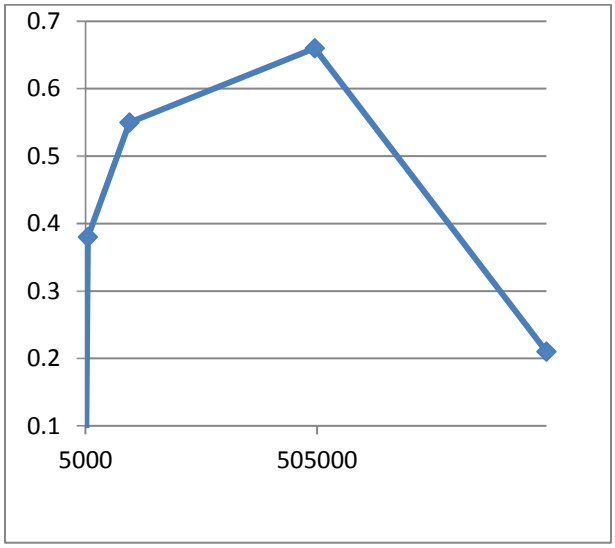

Figure 8: The p-value for Test for Longest Run of Ones in a Block (n= 6272-1000000)

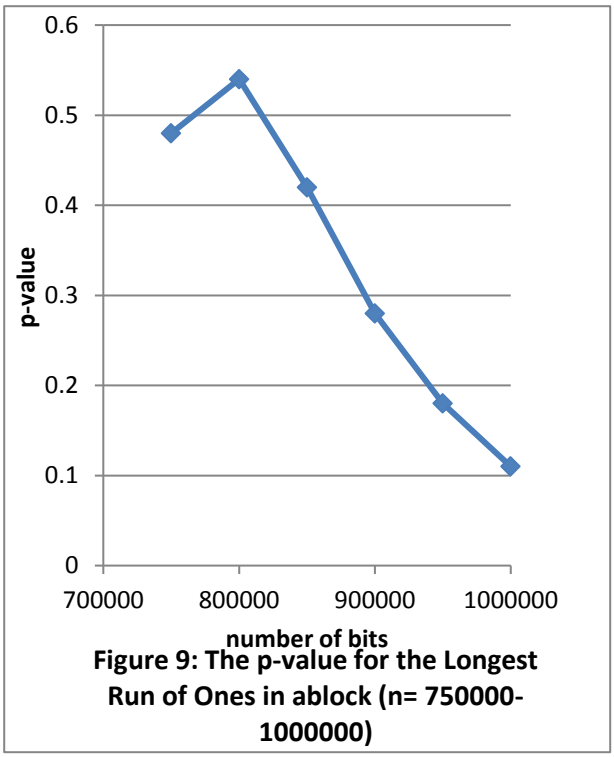

Figure 9: The p-value for the Longest Run of Ones in ablock (n= 750000-1000000)

4.1.5. Result for the Non-overlapping Template Matching Test.

In this test, we set the values for input size that it was recommended by the NIST. We generate the sequence of bits (n) to become 1048576 bits. And also we set the number of blocks (N) to 8 blocks, and we set the number of bits for each block to 131072 bits through the equation M= n/N. After that, we tested sequence of bits for this test and the result for the p-value was 0.9236 so the sequence is random because the p-value is more than 0.01.

4.2 Test the Speed of generating

In this section, the speed of generating sequences are very important when designing any generator key. It should be very high to make the encryption process take less time. So it was necessary known this test in order to determine the efficiency of our generator proposed design as shown in table 1. It was demonstrated a group of sequences that were distributed for seven sets (from 500-1000000 bits). when our design generated 500 bits, the time of execution it took 63 ms, so it was a good time. However, when the number of bits was increased into a double (1000-bits), the time of execution was 91 ms, then it considered an appropriate time. Moreover, when n was set to be 100000-bits, the time was 307 ms. In addition, when n was chosen to be $5* 10^5$, and 10^6 consecutively, the time was 1024 and 3755 ms. Nevertheless, the time of generating the sequences in all cases was perfect although there was a slight increase in execution time when the number of bits was increased.

Table1. Calculate the time of execution

No	No of bits (n)	Time of execution in (ms)
1	500	63
2	1000	91
3	5000	133
4	8000	140
5	100000	307
6	500000	1024
7	1000000	3755

4.3 Comparision our Generator Design with the Previous Designs

In this section, we will compare the results around our proposed design with some types of propose generators through measure the randomness level by using NIST tests as shown in Table 2. In paper [25], it talks about many techniques that used in the process of encryption for many communication systems. We will select one of them that called a logistic map. It consists of two types skew Tent Map (STM) and Modified Logistic Map (MLM). Both types contain of several properties which are the simpleness, high productivity, and fast characteristics. However, the lengths of sequence sequences resulting from STM and MLM are often much shorter than the other generators. So the resulting in the randomness sequences are very weak.

Table 2 demonstrates the results for the p-value, where it showed all values were low than 0.01 for the STM. Also, most p-values results for MLM were close to 0.001 but except one test (Nonoverlapping). It was equal to 0.9. Finally, we proved that our design pass all tests and it gives good randomness, very fast in implementation, and high level in security & complexity than SLM and MLM.

Table 2. Results of P-value for STM, and MLM with Our Design

No	Type of Test	STM	MLM	Our design
1	Monobit	0.001	0.001	0.59
2	Freq test within ablock	0.001	0.001	1
3	Runs	0.001	0.001	0.26
4	Longest Ones	0.001	0.001	0.21
5	Nonoverlapping	0.001	0.9	0.9236

5. Conclusion

In this study, a new design was proposed for the stream cipher through generation of Pseudo Random Number Generator (PRNG) which are used and considered appropriate in modern communication systems. This way consists of three types of generators that we have introduced into the combiner to convert them into a nonlinear Boolean function in order to make the key algorithm more secure. Through the obtained results and security analysis we can say that the proposed design obtained is very strong, and its' key is random and very hard to be broken. All these features demonstrate that our broadcast algorithm encryption is appropriate to encrypt data before transmission over public transport channels.

Future work

First of all, we are going to test the rest of the NIST tests to evaluate our design. In addition, we are goimg to create a new application for android mobile phone based on our design. Finally, we are planning also to encrypt voice and video by using our design.

CONFLICT OF INTERESTS.

- There are no conflicts of interest.

References

[1] O. Jallouli, .S. El Assad, and M. Chetto, "Robust Chaos-based Stream-Cipher for Secure Public Communication Channels," in *IEEE\ The 11th International Conference for Internet Technology and Secured Transactions*, vol. 73, 2016, pp. 23-26.

[2] C. Manifavas, G. Hatzivasilis, K. Fysarakis, and y. Papaefstathiou, "A survey of lightweight stream ciphers for embedded systems," *SECURITY AND COMMUNICATION NETWORKS*, vol. 9, pp. 1226–1246, 21 December 2016.

[3] M. Hamann, M. Krause, and W. Meier, "LIZARD – A Lightweight Stream Cipher for Power-constrained Devices," *IACR Trans Symmetric Cryptol*, pp. 45–79, 2017.

[4] G. Vidal, M. S. Baptista, and H. Mancini, "A fast and light stream cipher for smartphones," *The European Physical Journal Special Topics*, vol. 223, no. no.8, pp. 1601-1610, 13 May 2014.

[5] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Massachusetts, 1996.

[6] J. P. Aumasson, *Serious Cryptography: A Practical Introduction to Modern Encryption*.: No Starch Press, 2018.

[7] B. Schneier, *Applied Cryptography(20th ed.) Protocols, Algorithms and Source Code in C*. New york: Wiley, 2015.

[8] R.A. Rueppel, *Analysis and design of stream ciphers*.: Springer-Verlag, 1986.

[9] E. Dubrova, "A List of Maximum Period NLFSRs," 2012.

[10] L. Zhiqiang, "The Transformation from the Galois NLFSR to the fibonacci configuration," , 2013.

[11] "S. Golomb, Shift Register Sequences: Aegean Park Press, 1982.".

[12] "E. Dubrova,M. Teslenko, and H. Tenhunen, “On analysis and synthesis of (n, k)-nonlinear feedback shift registers,” in Proc. Design, Automation and Test in Europe Int. Conf., Munich, Germany, Mar. 2008, pp. 133-137".

[13] X.W. Wu, S. N. Koh, and C.C. Chui, "Primitive Polynomials for Robust Scramblers and Stream Ciphers Against Reverse Engineering," 2010.

[14] J. Massey, "Shift-Register Synthesis and BCH Decoding”, IEEE Trans. Inform. Theory,IT-15:122-127, January 1969".

[15] T. Brock, and R. Richard, "Linear Feedback Shift Register Sequences and the Berlekamp", Dec 10, 2005 ".

[16] "A. Klapper and M. Goresky, "Feedback Shift Registers, Combiners with Memory, and 2-Adic Span", J. Cryptology 10 (1997) 111-147".

[17] B. M. M. de Weger, "Approximation lattices of p-adic numbers", *J. Number Theory*, vol. 24, 1986, pp. 70–88.

[18] C. K. Wu and D. Feng , *Boolean Functions and Their Applications in Cryptography*.: Springer-Verlag Berlin Heidelberg, 2016.

[19] S. Fu, B. Sun, C. Li, and L. Qu, "Construction of Odd-Variable Resilient Boolean Functions with Optimal Degree", *Journal of Information Science and Engineering*, vol. 27, 2011.

[20] S. Pan , X. Fu , and W. Zhang, "Construction of 1-Resilient Boolean Functions with Optimal Algebraic Immunity and Good Nonlinearity", May 9, 2010.

[21] F. Zhang, Y. Hu, M. Xie, and Y. Wei, "Constructions of 1-resilient Boolean functions on odd number of variables with a high nonlinearity", *Security and Communication Networks*, vol. 5, pp. 614-624, 16 August 2011.

[22] L. Burnett, W. Millan, E. Dawson, and A. Clark, "Simpler methods for generating better Boolean functions with good cryptographic properties", *Australasian Journal of Combinatorics*, vol. 29, pp. 231-247, 2004.

[23] A. M. Inthair, T. S. Atia, and A. Y. Yousuf, "Design and Implement Stream Cipher System Using Saturated Best Resilient Function (SBR)", in *IEEE*, 2017.

[24] A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, A. Heckert, J. Dray, and S. Vo, "statistical test suite for random and pseudorandom number generators for cryptographic applications," *NIST Special Publication*, vol. 800-22, p. 131 pages, Aprli 2010.

[25] M. Garcia-Bosque, C. Sánchez-Azqueta, G. Royo, and S. Celma, "Lightweight ciphers based on chaotic Map - LFSR architectures", in *2016 12th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, Lisbon, 25 July 2016, pp. 1-4.

قوة التشفير الانسيابي لتأمين البيانات في الهواتف الذكية

عبدالله سعد عابد

طالب ماجستير، كلية الهندسية الكهربائية، هندسة تقنيات الحاسوب، الجامعة التقنية الوسطى

orjent_2007@yahoo.com

محمد جودة زعيتر

قسم هندسة تقنيات الحاسوب، كلية الهندسية الكهربائية، الجامعة التقنية الوسطى

mjzaiter@eetc.mtu.edu.iq

تيسير سلمان عطية

قسم هندسة الحاسوب، كلية الهندسة، الجامعة العراقية

tayseer_salman@yahoo.com

الخلاصة

مع التطور الحاصل في أجهزة الاتصالات وتكنولوجيا المعلومات وانتشارها بمساحات واسعة في العالم ادى ذلك الى زيادة المشاكل الامنية اثناء انتقال البيانات عبر تلك الاجهزة، على سبيل المثال تسرب البيانات، تعديلها، الوصول الغير مصرح به، و تعرضها الى الهجمات. توجد عدة انواع من التقنيات التي تستخدم لمنع حدوث هذه المشاكل وتعمل على حماية البيانات. أحدى هذه التقنيات هي التشفير الانسيابي التي تعتبر من اقوى واسرع الطرق المستخدمة في عملية التشفير وفك التشفير. في هذا البحث قدمنا تصميم جديد للتشفير الانسيابي لحماية بيانات الهاتف. قوة التشفير الانسيابي تعتمد على المفتاح. توجد عدة طرق تستخدم لغرض توليد المفتاح. استخدمنا ثلاثة انواع من المولدات(LFSR,FCSR,NLFSR) وبعد ذلك استخدمنا Non-linear Boolean function من نوع (f combination function) من نوع خمسة متغيرات، لجعل المولد اكثر اماناً. قمنا بدمج هذه الانواع الثلاثة وذلك بادخال أربعة مولدات من نوع LFSR مع احدى مولدات FCSR أو NLFSR الى f. بعد ذلك قمنا باختبار المفتاح من خلال قياس عشوائيته (التي تعتبر من اهم المعايير الاساسية لقياس كفاءة المولد) باستخدام اختبارات NIST statistical tests.

الكلمات الدالة: التشفير الانسيابي، المسجل الازاحة الخطي ذو التغذية الخلفية، المسجل الازاحة الحامل ذو التغذية الخلفية، المسجل الازاحة اللاخطي ذو التغذية الخلفية، دالة مركبة غير خطية، اختبارات احصائية ل.NIST