ISSN: 0067-2904

Design of New Efficient Stream Key Generator to Protect the Classified Information

Shahad Sameer Abed[1], Ayad Ghazi Nasir[2]
[1] Iraqi Commission for Computer and Informatics/Informatics Institute for Postgraduate Studies, Iraq, Baghdad
[2] Ministry of Education, General Directorate of Vocational Education, Baghdad, Iraq

Received: 24/1/2023 Accepted: 17/5/2023 Published: 30/5/2024

Abstract:
In this paper, we will create a new stream key cipher generator (KG) that is based on the LFSR unit and a chaotic map. The created KG can be used for a variety of purposes, including cryptography and steganography. The suggested KG is called Efficient Stream KG, and it demonstrates its efficiency when it passes all of the basic efficiency criteria tests, namely the periodicity, linear complexity, correlation immunity, and randomness.

Keywords: Stream ciphers، Linear Feedback Shift Registers، Key Generator, Chaotic Map.

مولد مفتاح انسيابي جديد كفوء لحماية المعلومات المصنفة

شهد سمير عبد العبيدي[*1] ، إيادغازي ناصر الشمري[2]
[1]الهيئة العراقية للحاسبات والمعلوماتية/معهد المعلوماتية للدراسات العليا، بغداد، العراق
[2]وزارة التربية، المديرية العامة للتعليم المهني ، بغداد، العراق

الخلاصة
في هذا البحث ، سيتم انشاء مولد تشفير مفتاح دفق جديد (KG) يعتمد على وحدة المسجل الزاحف الخطي ذو التغذية الخلفية ودالة فوضوية. يمكن استخدام KG الذي تم إنشاؤه لمجموعة متنوعة من الأغراض ، بما في ذلك التشفير وإخفاء المعلومات. يسمى KG المقترح Efficient Stream KG ، ويظهر كفاءته عندما يجتاز جميع اختبارات معايير الكفاءة الأساسية (الدورة ، التعقيد الخطي ، مناعة الارتباط ، والعشوائية)

1. Introduction

The internet has played an important role in communication and information sharing in today's information technology era. The rapid development of information technology, communication, and the internet has raised concerns about the security of data and information. Every day, confidential data is compromised, and unauthorized data access exceeds the limits. Great care should be taken to safeguard data and information. When combined with encryption, steganography will be a powerful and efficient tool that provides a high level of security[1] .

Many researchers are attempting to develop KGs for usage in a variety of domains, including cryptography, steganography, and computer games. In 2018 [2], Ali and Naser

*Email: ms202120665@iips.icci.edu.iq

introduced the high efficiency non-linear shift register generator, a new stream cipher key generator that meets all basic efficiency criteria. A new stream cipher is created as a clock-controlled one by Ashouri in 2018 [3], in his work, but with a novel system theory-based technique for altering steps that makes the structures used therein resistant to conventional assaults. Our proposed algorithm (PALS) uses a 32-bit message key in addition to a 256-bit main key. In 2018 [4], Ali and Naser came up with a new method using a stream cipher technique with dynamic issues. The mathematical model of the dynamic stream cipher method is based on the idea of changing the structure of the combined LFSRs with each change in the initial keys. In 2021, Murtaza et. al. [5], In their paper, developed and implemented an S-box generator that can be used in lightweight cryptography and outperforms previously designed S-box generators in terms of computation time and security resistance. To accomplish this, they used ordered Elliptic Curves (ECs) of small size and binary sequences to generate specific integer sequences, which are then used to generate S-boxes. Arockiasamy et. al. in 2021[6] , suggested leveraging tried-and-true generic concepts from contemporary cryptography to develop and implement a chaos-based cryptographically Secure Pseudorandom Number Generator (CSPRNG). They proposed a design and instantiation approach to chaos-based CSPRNG using proven generic constructions of modern cryptography. The proposed design approach with proper instantiation of such generic constructions eventually results in providing the best of both worlds that is the provable security guarantees of modern cryptography and passing of necessary statistical tests as that of chaos-based schemes. Also, they introduced a new coupled map lattice based on a logistic-sine map for the construction of CSPRNG. The proposed pseudorandom number generator is proven using rigorous security analysis as that of modern cryptography and tested using the standard statistical testing suites.

This paper is organized as follows: in section 2, we will discuss cryptography and the explanation of stream cipher and a feedback shift register. In section 3, the basic efficiency criteria for stream KG. The core principle of a chaotic map will be demonstrated in section 4. Section 5 shows the design of a new efficient stream KG(ESKG). The application of the BEC for ESKG is covered in section 6. Finally, in section 7, we will discuss some findings and future research.

2. Cryptography

Cryptography is the art of secret writing that has been used since Roman times to conceal information or to keep messages secure. Encryption/decryption is a popular method for keeping information private. Encryption and decryption are the fundamental functions of the cryptography. A simple message (plaintext) is converted into an unreadable form called ciphertext during encryption. A ciphertext is converted into the original text during decryption (plaintext). Both of these functions are used to protect messages from those who are not authorized to view their contents [7].

Symmetric and asymmetric cryptography are widely used types of cryptography, symmetric which is also known as symmetric key cryptography focuses on ensuring secure communication between sender and receiver by using the same secret key, whereas asymmetric is also known as public key cryptography secures communication by using public and private keys. Private keys are used for individual communication, whereas public keys are known to everyone due to their public nature. Figures 1 and 2 show symmetric and asymmetric cryptography, respectively. In both symmetric and asymmetric cryptography, the most crucial factor for communication security is the key size.

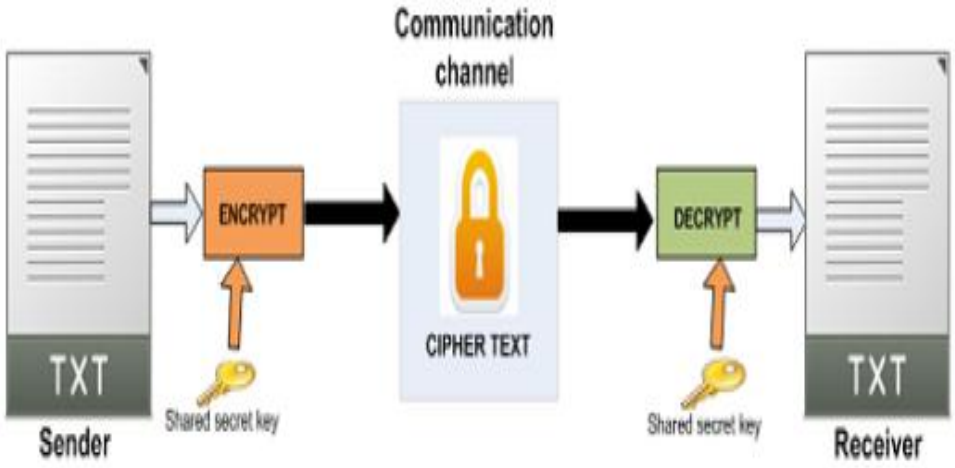

Figure 1: Symmetric Cryptography [7].

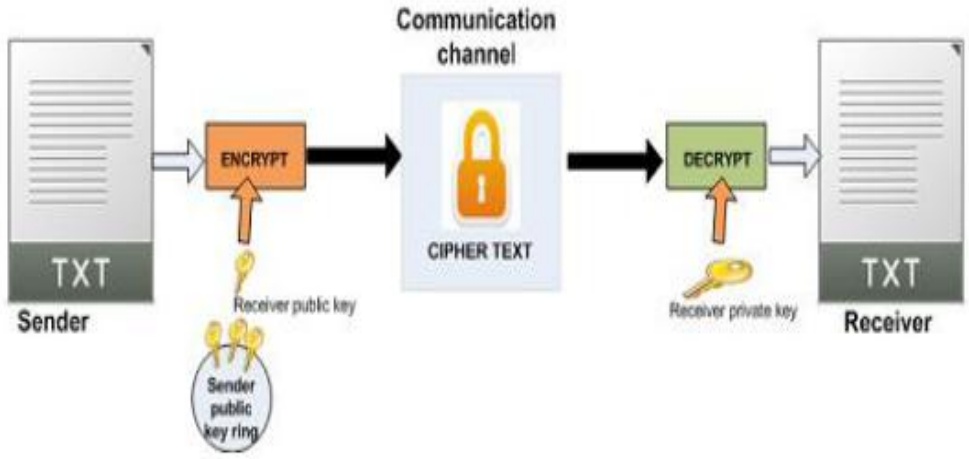

Figure 2: Asymmetric Cryptography [7].

2.1 Stream Cipher

A stream cipher is a kind of symmetric cryptosystem in which the plaintext is broken up into discrete units called characters and encrypted one character at a time before being bit-encoded. Bits serve as the message units in stream ciphers, and a random bit generator typically generates the key ,see Figure 3. Bit by bit, the plaintext is encrypted [2].

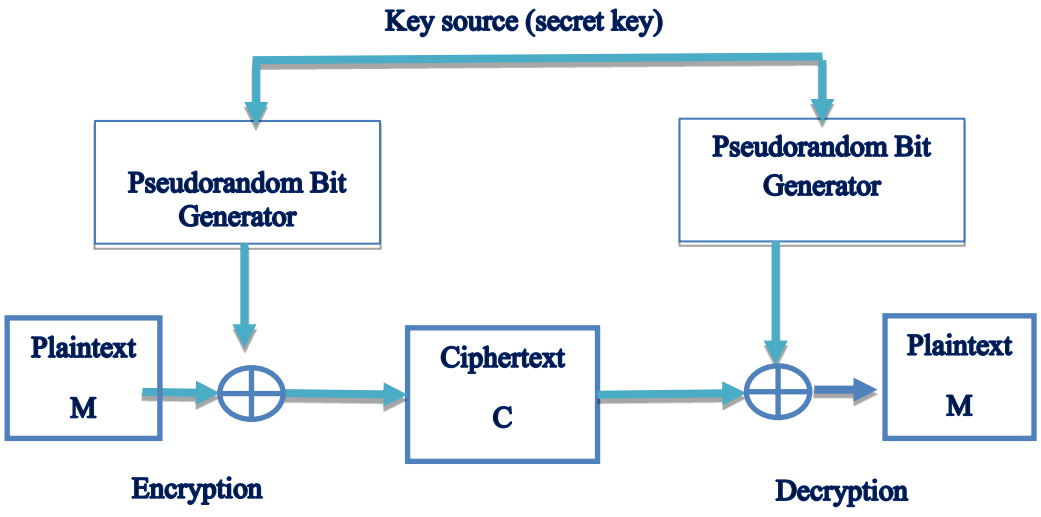

Figure 3: Stream Cipher System [2].

2.2 Linear Feedback Shift Register

One of the most frequent parts of the stream ciphers is the feedback shift register, which is classified as either Linear feedback shift register (LFSR) and Non-LFSR (NLFSR) depending on whether its feedback function is linear. Depending on the presentation mode, it is further divided into the Fibonacci pattern and the Galois pattern , see Figure 4. There are specific conversion relations [8].

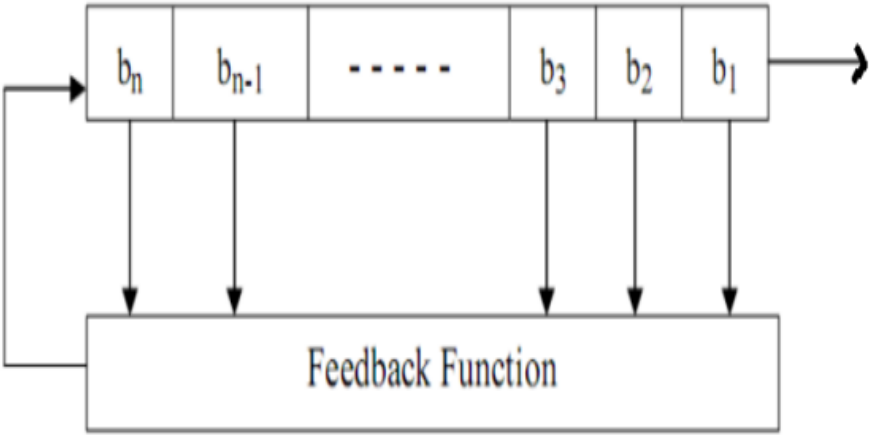

Figure 4: Linear Feedback Shift Register (LFSR) [9].

3. Basic Efficiency Criteria for Stream Key Generator

The first criterion for the KG efficiency is the sequence of the generator's key and its ability to resist the mathematical analysis that a cryptanalyst can apply to them[10]. The next subsections go through the basic efficiency criteria.

3.1 Periodicity

The length of the sequence S that the LFSR system generated will be denoted as P(S), and let the length of sequence S_i which is produced from $LFSR_i$ be $P(S_i)$ for $i = 1,2, \dots, N$, where N is the number of combined LFSRs. If $P(S_i) = 2^{r_i} - 1; r_i$ is the length of $LFSR_i$, which is calculated as follows:

$$P(S) = l.c.m(2^{r_1} - 1, 2^{r_2} - 1, \dots, 2^{r_N} - 1). \quad \dots (1)$$

If the greatest common divisor is the gcd between any two periods $2^{r_i} - 1$ and $2^{r_j} - 1, \forall\, i \neq j = 1, \dots, N$, gcd $(2^{ri} - 1, 2^{r_j} - 1)$=1, then $P(S) = \prod_{i=1}^{n}(2^{r_i} - 1)$[11].

3.2 Linear Complexity (LC) Criterion

A finite binary sequence's linear-complexity S is the length of the shortest LFSR that creates a sequence with S as one of its first terms, represented by $LC(S)$, which may be determined using the Berlekamp-Massey algorithm. The Berlekamp-Massey algorithm has the following steps for computing the linear complexity of a binary sequence [12].

Berlekamp-Massey Algorithm

Step(1): INPUT: A sequence of n elements of F_2 , $S^n = s_0, s_1, \dots, s_{n-1}$.

Step(2): OUTPUT: The linear complexity of S^n and $C(D)$ the feedback polynomial of length L of an LFSR which generates S^n.

Step(3):INITIALIZATION: $C(D) = 1, r = 0, m = -1, B(D) = 1, N = 0$.

Step(4): WHILE $(N < n)$ do the following:

Calculate the next difference, d: $d = \left(s_N + \sum_{i=1}^{r} c_i s_{(N-i)}\right) mod\ 2$.

IF $d = 1$ then do the following:
$T(D) = C\ (D), C(D) = C(D) + B(D).D^{N-m}$
IF $r \leq N/2$ THEN $r = N + 1 - r, m = N, B(D) = T(D)$.
$N = N + 1$.
ENDIF;
END WHILE
Step (5): END

3.3 Correlation Immunity (CI) Criterion

The correlation immunity is a link between the output sequence (S_i) of each combined LFSRi and the output sequence of the Combining Function Unit (CF) $CF = F_N$ generated from the KG. The correlation probability (CP) of x in general represents the ratio between the numbers of similar binaries (n_s) of two sequences to the length n of the compared part of them.

$$CP = n_s/n \quad \ldots (2)$$

If $0.45 \leq CP(x_i) \leq 0.55$, for $1 \leq i \leq m$ are statically independent of the output key (K), (where m is the number of immune LFSRs), then $CI = m$. We obtain the optimal value of the CI for any system when $m = N$ [13].

3.4 Randomness Criterion

The term random bit generator refers to a system or apparatus that can produce a series of binary digits that are statistically fair and independent (a sequence of bits). A pseudorandom bit generator (PRBG) generates another truly random binary sequence from a truly random digital sequence. The seed supports and outputs the contribution of the PRBG. The frequency, run, and autocorrelation tests are among the most important binary standard randomness tests[14].

A good PRBG must meet a number of statistical requirements such as the binary symbols must be balanced and the output symbols must be distributed evenly.
We will discuss the most five important randomness tests. Let S be the tested sequence of length n.

- **Frequency Test:** This test is used to assess how many 0s and 1s are in a sequence S (keystream sequence) of length n:

$$T_1 = \sum_{i=0}^{1} \frac{(n_i - n/2)^2}{n/2} = \frac{(n_0 - n_1)^2}{n}. \quad \ldots (3)$$

The observed number of n_i in S are indicated i's where $i = 0,1$. $E = n/2$ be the expected value of the event i, T_1distributed as chi-square distribution with one of freedom's degree [15].

- **Serial Test**: This test's objective is to determine whether the frequency of $00, 11, 10,$ and 01 as a subsequence S is close to the expected value for a random sequence:

$$T_2 = \sum_{i=0}^{1} \sum_{j=0}^{1} \frac{(n_{ij} - E)^2}{E}. \quad (4)$$

Where n_{ij} denotes the observed number of i and j bit in S, where $i, j = 0,1$, while the expected value $E = \frac{n-1}{4}$, and $\sum_{i=0}^{1} \sum_{j=0}^{1} n_{ij} = n - 1$, and $T_2 \sim \chi^2\ (0.05, 3)$.

- **Poker Test:** let n_i be the occurring frequency of the i^{th} type of a $m \geq 3$-length sequence. This test divides S into k m-long pieces. In order to assess whether a subsequence of length m occurs in S almost as frequently as a random sequence , so:

$T_3 = \sum_{i=0}^{m} \frac{(n_i - E_i)^2}{E_i}$. … (5)

Where $E_i = \binom{m}{i}(1/2^m)\ (n/m)$, and $T_3 \sim \chi^2\ (0.05, m)$ [16].

• **Run Test**: A run of S can be defined or known as many subsequences of sequence S which consists of concurrent or consecutive 0's or 1's. Blocks are the subsequences of 1s, whereas Gaps are the subsequences of 0s. By doing this test, you may find out if S has the expected amount of runs of different lengths from a random sequence. To put it in another way:

$T_4 = \sum_{i=1}^{k} \frac{(G_i - E_i)^2}{E_i} + \frac{(B_i - E_i)^2}{E_i}$. … (6)

We define G_i and B_i to be the occurrence number of gaps and blocks in terms of length i in S for each i; $1 \le i \le k$, where k is the largest gap or block that happened in the sequence S. The expected value $E_i = (n - i + 3)/2^{i+2}$ and $T_4 \sim \chi^2\ (0.05, 2k - 2)$.

• **Autocorrelation Test**: In this test, the sequence S and its (non-cyclic) shifted reversions are compared for similarity. Assume τ (number of shifting) is a fixed integer $1 \le \tau \le n/2$.

$T_5 = (n_0\ (\tau) - n_1\ (\tau))^2\ /\ (n - \tau)$. … (7)

See that $T_5 \sim \chi^2\ (0.05, 1)$ and the observed number of $n_i(\tau)$ in S are indicated the i's where $i = 0, 1$ in the shifted sequence. In each shift τ the length of S will be $n - \tau$ [17].

4. Chaotic Map

Chaos theory is an area of mathematics that deals with nonlinear dynamical systems. A system is nothing more than a group of linked parts that interact to create a bigger whole. The system is nonlinear as a result of feedback or multiplicative effects between the parts. Last but not least, the term dynamical suggests that the system changes throughout time in response to its current state. Almost every non-trivial real-world system is a nonlinear dynamical system. An example of a chaotic system is a nonlinear dynamical system, which can contain a few interconnected parts and adhere to straightforward laws but is always highly sensitive to its initial conditions [18]. The chaos theory has been already applied to a high-speed search. In addition, chaotic maps have become a popular topic in recent years as they are used in a variety of sectors to provide safe communication.

Many different types of chaos maps have been investigated in recent years. However, we will focus on the Gauss chaos map in this study. This map is known by several names, including the Gauss map and the Gaussian maps or mouse maps. It is a nonlinear iterated map of the real to a real range with the following form [19]:

$x_{n+1} = \beta \exp(-\alpha x_n^2)$ n=0,1, 2, … … (8)

When α and β are real parameters, the function map or the bell shape Gaussian functions which are similar to the logistic map with

$\alpha = 4.9$ and $\beta \in [-1,1]$ (9)

5. Design of New Stream Key Generator

In this study, we will create a new stream generator which is known as the Efficient Stream Key Generator (ESKG). This cryptosystem is considered a nonlinear system because of the Random Access Memory (RAM) unit so it is hard to be analyzed.

5.1 Key Management of ESKG

In this paper, we will suggest using two kinds of keys that are intended to be used as an initial key for the ESKG. These keys are:

1. **Initial Basic Key (IBK)**: This key is changed with each message and requires an essential private key that consists of (20) ASCII CODE (8 bits) characters. This key must be transmitted over a secure channel.

2. **Initial real x_0:** x_0 (which consists of 16 decimal numbers) for the Chaotic Map mentioned in relation (8).

5.2 ESKG Components

1. The Initial System (IS) LFSR's consists of 4 LFSR's, with the following characteristic polynomials:
a. LFSR1 has characteristic polynomial $x^{10} + x^3 + 1 \in GF(2)$.
b. LFSR2 has characteristic polynomial $x^{11} + x^2 + 1 \in GF(2)$.
c. LFSR3 has characteristic polynomial $x^{22} + x + 1 \in GF(2)$.
d. LFSR4 has characteristic polynomial $x^{25} + x^3 + 1 \in GF(2)$.
2. LFSR'S unit (LFSRU): this unit consists of 8 LFSRs with the following characteristic polynomials:
a. LFSR1 has characteristic polynomial $x^{28} + x^3 + 1 \in GF(2)$.
b. LFSR2 has characteristic polynomial $x^{31} + x^6 + 1 \in GF(2)$.
c. LFSR3 has characteristic polynomial $x^{35} + x^2 + 1 \in GF(2)$.
d. LFSR4 has characteristic polynomial $x^{39} + x^4 + 1 \in GF(2)$.
e. LFSR5 has characteristic polynomial $x^{41} + x^3 + 1 \in GF(2)$.
f. LFSR6 has characteristic polynomial $x^{49} + x^9 + 1 \in GF(2)$.
g. LFSR7 has characteristic polynomial $x^{57} + x^7 + 1 \in GF(2)$.
h. LFSR8 has characteristic polynomial $x^{60} + x^1 + 1 \in GF(2)$.
3. RAM Unit (RAMU): It consists of 256 random and different bytes.
4. Chaotic Map (Gauss Map (GM)), see relations (8) and (9).

5.3 ESKG Initialization

1. The IBK converts to 160 bits which are called BBK, so we have $BBK_k, i = 1, \dots, 20$; $j = 0, \dots, 7$; where $k = (i - 1) * 8 + j = 1,2, \dots, 160$, (see Table (1)).

Table 1: Bits of BK

I	1,1,…1	2,2,…2	...	20,20,…20
J	$0,1, \dots, 7$	$0,1, \dots, 7$	...	$0, 1, \dots, 7$
K	$1,2, \dots, 8$	$9,10, \dots, 16$	...	$153,154, ., 160$

2. The $BBK_k, k = 1,2, \dots, 160$, array fills the IS, and the last stages of each shift register are filled by 1.
3. The IS moves to fill the LFSRU system and the last stages of each shift register are filled by 1.
4. IS moves to generate 256 different that are not repeated and random bytes for RAMU.
5. The LFSRU moves to generate an address to RAMU to obtain byte (8 bits) to fill the BLFSR, the last stage is filled by 1.
6. The IS moves again to generate 256 distinct bytes to fill the RAMU, this byte is generated by the (4) output and (4) fixed positions (13, 17, 23 and 45) from each LFSR in IS.

5.4 ESKG Moving

1. The LFSRU moves to generate an address to the RAMU to get BY1 as output of the RAMU.
2. The GM moves one step through the initial x_0 to get x_1=BY2 through the relation (8). Therefore, the final key is:

KB=BY1 XOR BY2 ….(10)

and so on, repeating the work to generate bytes to be the final key for the ESKG.
Figure 5 shows the ESKG moving.

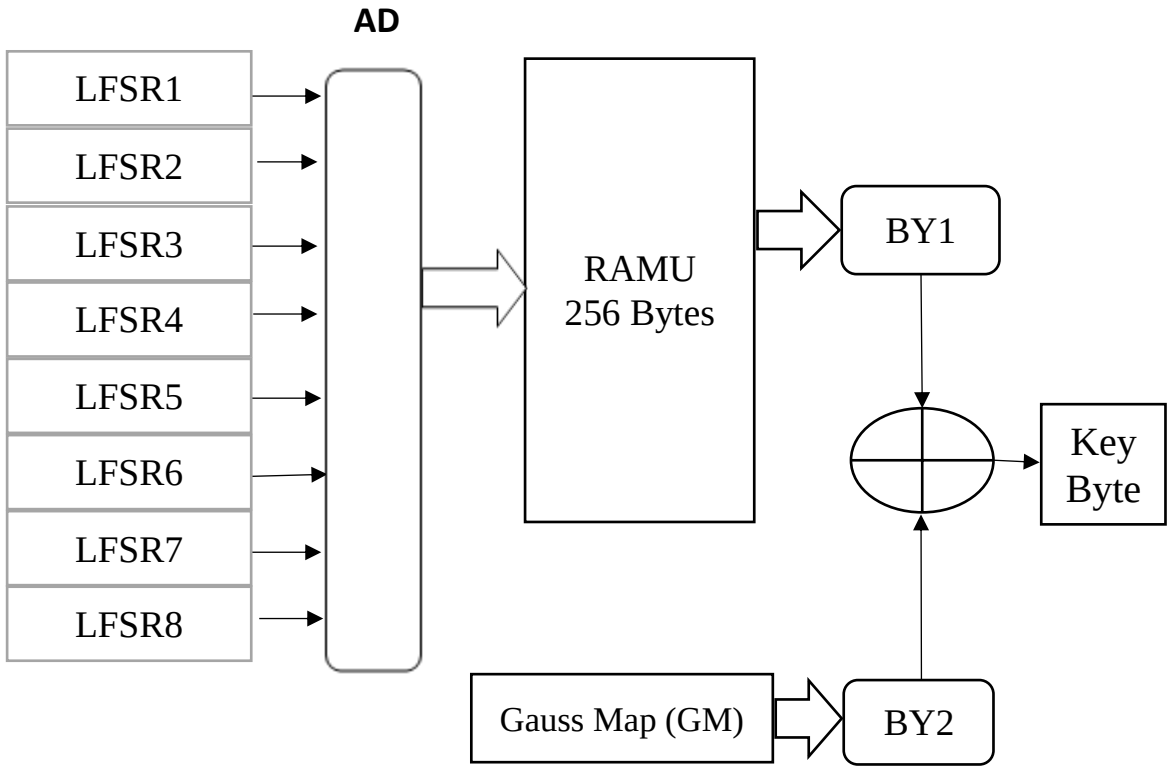

Figure 5: The block diagram of ESKG.

6. Applying of BEC on New Key generator

In this section, we will generate three examples of Key1, Key2 and Key3 of output keys from the ESKG with different lengths L_k=1000, 5000, and 10000 bits, respectively. These examples are tested with each test of BEC. Firstly, let S be the sequence that is generated from the generator ESKG and S_i be the sequence that is generated from $LFSR_i, i = 1,2, \dots,8$.

Periodicity ($P(S)$): We can calculate the $P(S)$ by using relation (1) as follows:

$$P(S) = l.c.m(2^{28} - 1, 2^{31} - 1, \dots, 2^{60} - 1). \quad \dots (11)$$

Relation (11) shows the high periodicity of the ESKG which is near to 6.3×10^{54}. It is important to mention that the periodicity does not depend on the combining function (RAMU).

Linear Complexity ($LC(S)$): In Table (2), we will show the length of equivalent LFSR to ESKG which means the $LC(S)$ of the three examples by applying the Berlekamp-Massey Algorithm.

Table 2: The $LC(S)$ of the three examples is for ESKG only.

Examples	L_k (bits)	$LC(S)$	Decision
Key1	1000	505	P
Key2	5000	2515	P
Key3	10000	5025	P

where P means pass and F means fail.

From Table (2), since $LC(S) \geq L_k/2, k = 1,2,3$ that means the ESKG has high linear complexity.

Correlation Immunity (CI): In Table (3), we will show the results of CP of applying the CI test as in relation (2) for the ESKG for the three examples.

Examples	x_j	1	2	3	4	5	6	7	8	CI
Key1	$CP(x_j)$	0.49	0.48	0.53	0.45	0.50	0.50	0.51	0.47	8
	Decision	P	P	P	P	P	P	P	P	P
Key2	$CP(x_j)$	0.46	0.51	0.52	0.53	0.50	0.52	0.47	0.52	8
	Decision	P	P	P	P	P	P	P	P	P
Key3	$CP(x_j)$	0.51	0.47	0.48	0.50	0.48	0.51	0.53	0.48	8
	Decision	P	P	P	P	P	P	P	P	P

In Table (3), we see that $0.45 \leq CP(x_j) \leq 0.55$ for $j = 1,2,\dots,8$, which means the ESKG is correlated immune since $m = 8$ for all outputs.

Randomness

In this subsection, we will apply the randomness tests for the three examples. In Table (4), the frequency test is applied on the output key of the ESKG to obtain the T value for the three examples using freedom degree $\nu = 1$ and $T_0 = 3.841$ which are obtained from χ^2 test using relation (3).

Table 4: Frequency Test Results for the ESKG.

Examples	T	Decision
Key1	0.001	P
Key 2	1.008	P
Key 3	0.462	P

In Table (5), the serial test is applied on the output key of the ESKG to obtain the T value for the three examples using $\nu = 3$ and $T_0 = 7.815$ which are obtained from χ^2 test using relation (4).

Table 5: Serial Test Results for ESKG.

Examples	T	Decision
Key1	1.648	P
Key 2	1.865	P
Key 3	0.482	P

In Table (6), the results of the Run test that is applied on the output key of ESKG to obtain the T value for the three examples using different ν and T_0 which are obtained from χ^2 test using relation(5).

Table 6: Run Test Results for the ESKG

Examples	T	ν	T_0	Decision
Key1	8.652	8	13.362	P
Key 2	8.206	11	17.275	P
Key 3	9.538	9	16.919	P

Poker test results are shown in Table (7) for the output key of the ESKG to obtain the T value for the three examples using different $\nu = 3$ and $T_0 = 7.815$ which are obtained from χ^2 test using relation (6).

Table 7: Poker Test Results for ESKG.

Examples	T	Decision
Key1	7.344	P
Key 2	1.273	P
Key 3	7.112	P

In Table (8), the Auto-correlation test is applied on the output key of the ESKG to obtain the T value for the three examples using $\nu = 1$ and $T_0 = 3.841$ which are obtained from χ^2 test using relation (7) for $= 1,2, \dots ,10$.

Table 8: Auto-correlation test results for the ESKG.

τ	Key1		Key2		Key3	
	T	Decision	T	Decision	T	Decision
1	1.523	P	0.135	P	0.533	P
2	0.677	P	1.961	P	0.608	P
3	0.627	P	1.693	P	2.201	P
4	1.028	P	0.016	P	0.640	P
5	0.443	P	0.353	P	0.008	P
6	0.402	P	0.370	P	0.010	P
7	1.862	P	0.115	P	0.029	P
8	2.133	P	3.859	F	2.373	P
9	0.849	P	1.413	P	0.423	P
10	2.525	P	1.517	P	3.100	P

Tables (4-8) show the high randomness of the ESKG, where the five randomness tests Frequency, Serial, Run, Poker, and Auto-correlation are applied where the output key passes all the randomness efficiently.

7. Implementation System of the ESKG

The ESKG system was tested by the programs Visual Studio 2013 version 13.0 and Processor Intel® Core(TM) i3 CPU,2.53 GHz, Core(s), with Ram 1.21 GB computer.

8. Conclusions and Future Works

1. The constructed ESKG has a high non-linearity which gives a high linear complexity for the designed stream KG.
2. We avoid the correlation which happened because of the high non-linearity by using the GM which is considered a balanced part.
3. All the BEC test results prove the efficiency of the ESKG when it passes all these tests.
4. The suggested ESKG can be developed by increasing the number of the LFSRs or their lengths and increasing the size of the RAM units, all those suggestions will be contributed to increase the complexity.
5. Since our proposed ESKG passes all the tests, we recommend it to be used in protecting the data with many applications like cryptography and steganography.

References

[1] Y. A. Yunus, S. A. Rahman, and J. Ibrahim, "Steganography: A Review of Information Security Research and Development in Muslim World," *Am. J. Eng. Res.*, vol. 02, no. 11, pp. 122–128, 2013, [Online]. Available: www.ajer.org.

[2] A. A. Ghazi and F. H. Ali, "Design of New Dynamic Cryptosystem with High Software Protection," *Iraqi J. Sci.*, pp. 2301–2309, 2018.

[3] Ashouri ,"New Efficient Stream Key Generator to Protect the Classified Information Ayad Ghazi Nasir2," pp. 1–14, 2018.

[4] A. A. Ghazi and F. H. Ali, "Robust and efficient dynamic stream cipher cryptosystem," *Iraqi J. Sci.*, vol. 59, no. 2, pp. 1105–1114, 2018, doi: 10.24996/IJS.2018.59.2C.15.

[5] G. Murtaza, N. A. Azam, and U. Hayat, "Designing an Efficient and Highly Dynamic Substitution-Box Generator for Block Ciphers Based on Finite Elliptic Curves," *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/3367521.

[6] J. P. Arockiasamy, L. E. Benjamin, and R. U. Vaidyanathan, "Beyond Statistical Analysis in Chaos-Based CSPRNG Design," *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/5597720.

[7] F. Maqsood, M. Ahmed, M. Mumtaz, and M. Ali, "Cryptography: A Comparative Analysis for Modern Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 442–448, 2017, doi: 10.14569/ijacsa.2017.080659.

[8] L. Jiao, Y. Hao, and D. Feng, "Stream cipher designs: a review," *Sci. China Inf. Sci.*, vol. 63, no. 3, pp. 1–25, 2020, doi: 10.1007/s11432-018-9929-x.

[9] A. Negi, J. S. Farswan, V. M. Thakkar, and S. Ghansala, "Cryptography playfair cipher using linear feedback shift register," *IOSR J. Eng.*, vol. 2, no. 5, pp. 1212–1216, 2012.

[10] A. G. Nasser, "New Design of Efficient Non-Linear Stream Key Generator," no. 1, pp. 1–15.

[11] P. P. Deepthi and P. S. Sathidevi, "Design, implementation and analysis of hardware efficient stream ciphers using LFSR based hash functions," *Comput. Secur.*, vol. 28, no. 3–4, pp. 229–241, 2009, doi: 10.1016/j.cose.2008.11.006.

[12] A. A. Alsaadi and A. G. Naser Al-Shammari, "Enhancement of non-linear generators and calculate the randomness test for autocorrelation property," *Iraqi J. Sci.*, vol. 60, no. 10, pp. 2229–2236, 2019, doi: 10.24996/ijs.2019.60.10.17.

[13] H. A. M. Al_Sharifi, "Frequency Postulate's Theoretical Calculation for the Sequences Produced by Modified Geffe Generator," *J. kerbala Univ.*, vol. 12, no. 2, 2014.

[14] G. Gong, T. Helleseth, and P. V. Kumar, "Solomon W. Golomb - Mathematician, Engineer, and Pioneer," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2844–2857, 2018, doi: 10.1109/TIT.2018.2809497.

[15] A. S. Babu and B. Anand, "Modified dynamic current mode logic based LFSR for low power applications," *Microprocess. Microsyst.*, vol. 72, p. 102945, 2020, doi: 10.1016/j.micpro.2019.102945.

[16] S. V. Sathyanarayana, M. Aswatha Kumar, and K. N. Hari Bhat, "Random binary and non-binary sequences derived from random sequence of points on cyclic elliptic curve over finite field GF(2m) and their properties," *Inf. Secur. J.*, vol. 19, no. 2, pp. 84–94, 2010, doi: 10.1080/19393550903482759.

[17] K. B. Sudeepa, G. Aithal, V. Rajinikanth, and S. C. Satapathy, "Genetic algorithm based key sequence generation for cipher system," *Pattern Recognit. Lett.*, vol. 133, pp. 341–348, 2020, doi: 10.1016/j.patrec.2020.03.015.

[18] Y. Tang, J. Kurths, W. Lin, E. Ott, and L. Kocarev, "Introduction to focus issue: When machine learning meets complex systems: Networks, chaos, and nonlinear dynamics," *Chaos An Interdiscip. J. Nonlinear Sci.*, vol. 30, no. 6, p. 63151, 2020.

[19] K. M. Ali and M. Khan, "Application Based Construction and Optimization of Substitution Boxes Over 2D Mixed Chaotic Maps," *Int. J. Theor. Phys.*, vol. 58, no. 9, pp. 3091–3117, 2019, doi: 10.1007/s10773-019-04188-3.