IHJPAS. 37 (1) 2024
Ibn Al-Haitham Journal for Pure and Applied Sciences
Journal homepage: jih.uobaghdad.edu.iq
PISSN: 1609-4042, EISSN: 2521-3407

Hybrid Efficient Stream Cipher KeyGenerator Based on LFSR's and Chaotic Map

[1]Dina.H.Abbas* [2]Ayad Abd-al-Kahhar AbdulSalam
[1,2]Department of Computer, College of Sciences for Women, University of Baghdad, Baghdad, Iraq.

*Corresponding Author. dina.h@csw.uobaghdad.edu.iq

Received: 9 March 2023, Received 1 April 2023, Accepted 9 April 2023, Published 20 January 2024

doi.org/10.30526/37.1.3321

Abstract

Communication security that depends on chaos can be considered as a new approach which provides protection and security of communications and maintains confidentiality because Chaos theory can be implemented in cryptosystem successfully.

A stream cipher, on the other hand, is a type of symmetric cryptosystem in which the plaintext is divided into small entities known as characters. The key in stream cipher is typically generated by a random bit generator. Many key stream generators employ linear feedback shift registers (LFSRs). LFSR systems are made up of a group of LFSR units and a combining function (CF) unit. The plaintext is encrypted one bit at a time. The key is fed into a random bit generator, which produces a long series of binary signals. This "key-stream" k is then combined with plaintext m, typically via a bit-wise XOR (Exclusive-OR modulo 2 addition), to produce the ciphertext stream, which employs the same random bit generator and seed.

In this paper we will introduce a new stream cipher keygenerator which using a hybrid between chaotic and combination of Linear Feedback Shift Registers (LFSR's). The proposed generator can be used to protect different types of data files (text, image, audio and video). Many kinds of tests are applied to specifying the goodness of the proposed keygenerator. The results of testing prove the efficiency of the suggested system.

Keywords Cryptography, Stream Cipher, Chaotic map, Linear Feedback Shift Registers, Randomness tests.

1. Introduction

Cryptology is the science which studies cryptography and cryptanalysis. Cryptography is the science which is concerned with designing cryptosystems to encrypt messages and decrypt it. And Cryptanalysis, deals with encryption and encrypted messages, hoping to find hidden messages, without prior detailed knowledge of the cryptosystem [1].

In order to keep sensitive information secure from public scrutiny, mathematicians have developed a field known as cryptography to research and develop methods for communicating data in a protected form, even if the transmission is done through an insecure channel it can be defined as a process of protecting data by manipulating it in some way using various mathematical operations and keys to lock and then unlock the information, so that it can only be interpreted and processed by the intended sender and receiver, Thus, preventing the unauthorized access to information, because the techniques used to encrypt/decrypt information in cryptography are made from mathematical principles and algorithms that translate messages in ways that make it difficult to discern and decode [2].

The sender (Alice) wants to send the original form of information which is called Plaintext (P) message to the receiver (Bob) through a secure or insecure communication channel by uses the crypto tool. Trudy (T), an outsider, has intentions of listening in on the message. The sender uses cryptographic methods to disguise the message for security purposes. Encryption (E) is the final step in the process of hiding information using an algorithm, and it requires more complex mathematical operations. Encryption is the technique by which a communication is made unreadable to anybody but the recipient by employing an encryption key (Ke), and the encrypted message itself is called cipher text (C). Decryption (D) is the inverse of encryption, where P is extracted from C using the decryption key (Kd). Below are some equations illustrating the mathematical model of the above processes[3]:

$$C = E(P, Ke) \quad (1)$$

$$P = D(C, Kd) \quad (2)$$

Cryptosystems classified according to types of key to symmetric and asymmetric cryptography. In symmetric cryptography we use the same key for encryption and decryption, while in asymmetric cryptography (also known as public key cryptography) we use two types of keys, one for encryption and the other for decryption. Symmetric key cryptography contains two types of cipher, stream cipher and block cipher [4].

Chaos theory is a branch of mathematics and physical studies. It deals with the behavior of nonlinear dynamic systems. In its applications, contemporary cryptography makes use of a chaotic system. Desirable characteristics of chaos include determinism, nonlinearity, irregularity, long-term prediction, and sensitivity to beginning conditions. These characteristics have drawn numerous researchers to employ chaos in contemporary cryptography. The chaotic system is a tool used in modern cryptography.

In [5], In that paper, a new cryptographic method was proposed, designed for audio files' security. The encryption algorithm was based on classic symmetric models using pseudo-random number generator composed with chaotic circle map and modified rotation equations. The results proved the high level of security, provided extensive cryptographic analysis were included key sensitivity analysis, key-space analysis, waveform and spectrogram analysis, correlation analysis, number of sample change rate analysis, level of noise analysis and speed performance test.

In [6], that study introduced a discrete modified Henon map-based audio encryption technique that was substituted with a key stream produced by a modified Lorenz-Hyperchaotic

system. In that study, the Fast Walsh Hadamard Transform (FWHT) was used to compress the audio file initially in order to remove any remaining intelligibility in the transform domain. A mechanism for dynamic key stream generation was used to improve the correlation between plaintext.

In [7], the researcher was used the chaotic system in encryption because chaotic sensitivity to initial conditions, control parameters, and pseudo-randomness. Its crypto-algorithms were appropriate for extensive data encryption for images, audio or videos. In that paper, an audio encryption algorithm based on substituted and permutated network was proposed. In that study, Mobius transformation had been used as a source to generate strong S-boxes for the substitution network and the Hénon chaotic map that performed a pixel-wise permutation as employed for the permutation network. The algorithm had been tested several times with different sizes of audio files; the experimental outcomes showed that the researcher algorithm had been effective complexity and had suitable for audio encryption and hence for secure audio communication.

2. Stream Cipher [8]

A stream cipher is a sort of symmetric cryptosystem in which the plaintext is separated into little entities known as characters and encrypted bit by bit, character by character. Bits are used as the basic unit of communication in stream ciphers, with the key often being generated by a random bit generator . A bit-by-bit encryption process is used to protect the plaintext.

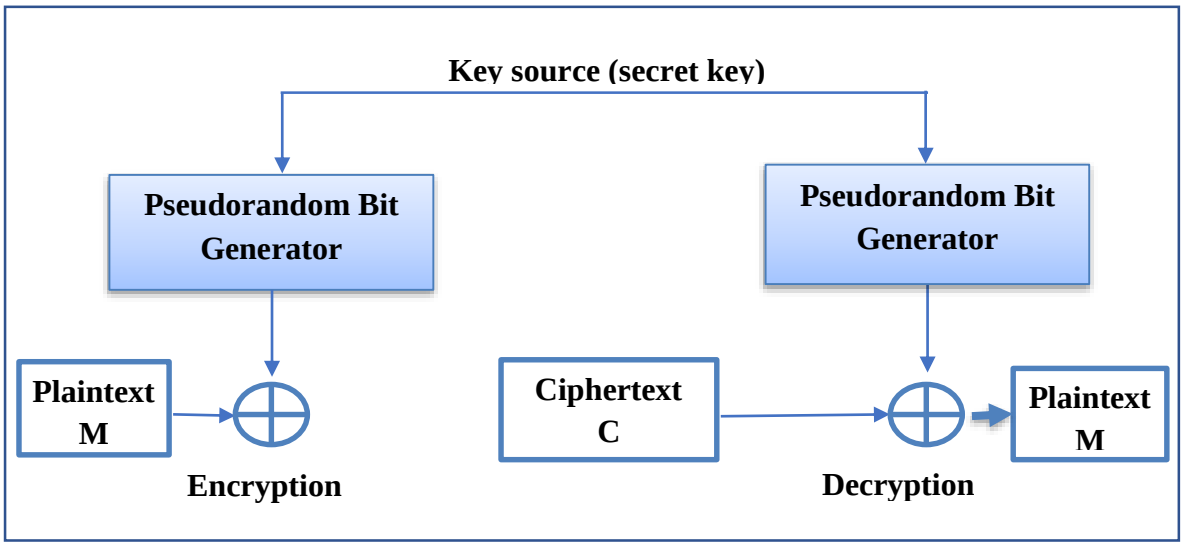

Figure 1. Stream Cipher System.

With the help of the key, a random bit generator (RBG) produces a long sequence of binary signals. Using the same random bit generator and seed, a "key-stream" k is generated, and it is combined with plaintext m to produce the ciphertext stream by means of a bit-wise XOR (Exclusive-OR modulo 2) addition.

Stream ciphers based on the shift register have been favored by cryptographers due to their simple implementation in digital hardware. Selme, the main cryptographer for the Norwegian government, developed the notion of SR sequences in 1965 [9]. Golomb, a mathematician at the NSA, published a book containing Selmers' findings and his own [10].

A shift register (SR) and its feedback function are the two primary components of a feedback shift register (FSR). It's a string of bits that makes up the SR (the length of a SR is figured in bits). Bits in the SR are shifted right by one position whenever only one bit is required. A Linear Feedback Shift Register (LFSR) is the simplest FSR. The register stages involved in the feedback function (polynomial) are merely XORed together. Due to the simple nature of the feedback sequence, a wide variety of mathematical tools may be used to analyze LFSRs.

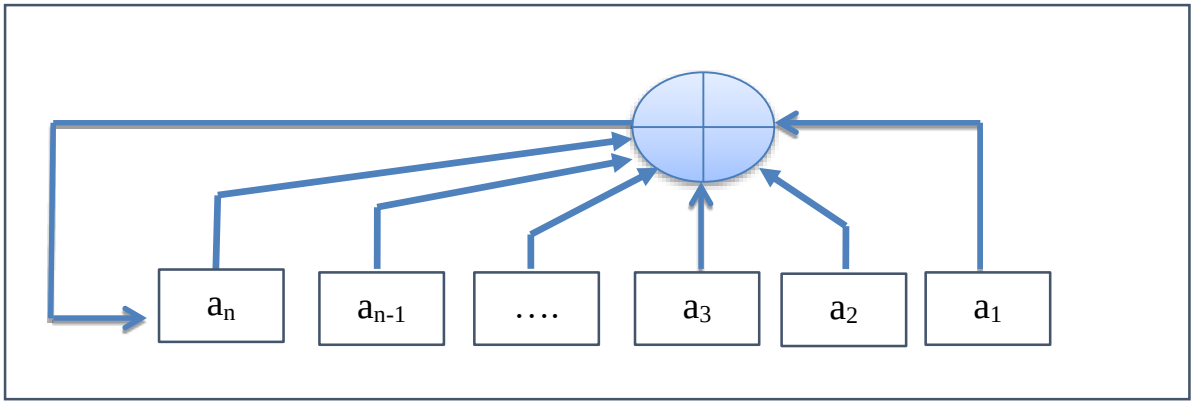

Figure 2. Linear Feedback Shift Register (LFSR).

Many of the most important suggested stream generators use linear feedback shift registers (LFSRs). For a variety of reasons [11]:

1. LFSRs are easy to implement in hardware.
2. They can generate long-period sequences.
3. They can generate sequences with useful statistical characteristics.
4. As a result of their structure, they can be easily analyzed using algebraic methods.

The basic units of LFSRs systems are consists of collection of LFSR units and combining function (CF) unit. The description of these units is as follows [12]:

1-LFSRs unit: Every LFSRs system contains a collection of linear shift registers, every one shifted alone in one time. As the nature of connection function, each LFSR produce a sequence can be described as independent sequence and LFSR unit depends on the following elements:

a-The lengths of each LFSR.
b-The connection function.
c-The initial value of LFSR.

2-Combining Function Unit: The combining function for n inputs denoted by CF_n is a Boolean function on Galois field GF (2) it's input are the sequence generated from each LFSR. In general If $x_1, x_2 \dots x_n$ are inputs of CF_n such that:

$x_i \in GF(2); i = 1, \dots, n$, Then

$$F_n(x_1, x_2 \dots x_n) = a_0 \oplus \sum_{i=1}^{n} a_i\ x_i \oplus \sum_{i,j}^{n} a_{ij}\ x_i \oplus \dots \oplus a_{1,2,\dots,n} \prod_{i=1}^{n} x_i \qquad (3)$$

Where a_0 , a_i , a_{ij}, … , $a_{1,2,\ldots,n} \in GF(2)$,the coefficients for the combining of LFSR's system.

3. Chaotic Map

As Chaos theory may be successfully implemented in cryptosystems, communication security that relies on it can be seen as a novel technique that protects communications, keeps them secure, and preserves their confidentiality. The field of mathematics and physics known as chaos theory. Nonlinear dynamic systems and their behaviors are the focus here. Deterministic, Irregular, Long-Term Prediction, Nonlinear, and Sensitivity to Initial Conditions are all desirable properties of chaos. According to the deterministic property, the future state of the chaos function is always dependent on the state before it. The behavior of a chaotic system is characterized by its irregular continuity, which is demonstrated by the presence of the irregular attribute. Because of this nonlinearity, the chaos function can undergo nonlinear transformations. As a result of the sensitivity to beginning conditions property, tiny changes in the initial state of chaotic systems can result in radically different final-state behavior. Predicting a system's behavior over a lengthy period of time is challenging because of the difficulties inherent in achieving features that are irregular and sensitive to initial conditions [13].

Because of these qualities, many researchers are interested in using chaos in modern Cryptography. In its applications, modern cryptography utilizes a chaotic system. Researchers have spent the last decade exploring various chaos-based cryptography methods, such as chaos-based block/stream cipher, chaos-based secret communication, chaos-based random number generation, chaos-based hash, etc.[14].

The following example illustrates the behavior of chaos. This example is intended to demonstrate that chaos is a dynamic system with a high sensitivity.

As known, chaos is extremely sensitive to its initial state x_n. If you like, you might think of the number sequences that chaos generates as secret keys that need to be guarded by two people that need to communicate with each other.

Some types of chaos maps used in chaos-based cryptography [15]:

1. Chaos and the Logistic Map.
2. Chaos and the Standard Map:
3. Chaos and the Piecewise Linear Chaotic Map (PLCM)
4. Chaos and the Lorenz Map
5. Chaos and the Henon Map
6. The chaos system that used in this paper are presented as follow:

$$\left.\begin{aligned} x &= x + (-s * x + y * k - r * p) \\ y &= y + (-y - x * z + r * x - u * p) \\ z &= z + (z * x * y - 1.5 * s * p - k) \\ k &= k + (s * x + (u * y - r * k) \\ p &= p + (b * ((x + k)/z) + y \end{aligned}\right\} \quad (4)$$

In which the system's behavior is represented by the vectors x, y, z, k, and p. the system exhibits chaotic behavior with $s = 0.95, r = 0.5, b = 0.01, u = 1.1$ and (x_0,y_0, z_0, p_0, k_0) are in the interval [0,1] [16].

4. Basic Efficiency Criteria for Cryptosystem

There are two fundamental parts to any stream key generator (SKG): the bit stream sequence S and the combining function F for the key generator. To ensure the integrity of the generated key sequence, the key generator must meet certain requirements before being built before it is constructed.

Here we will discuss the criteria for determining whether or not a sequence is efficient enough to be used as an encryption (or decryption) key. An efficient key generator is also compatible with an efficient sequence, and vice versa. These requirements for the efficiency of a key generator are dependent on some or all of the key generator's fundamental units; hence, they may overlap.

The basic efficiency for the cryptosystem can be defined as the ability of the generator and its sequence hold one's own the mathematical analytic which the cryptanalyst applied on them. This ability measured by some basic criteria to test the key generator efficiency [17]. SKG should have:

1. Have large linear complexity.
2. Have maximum period.
3. Be a correlation immune.
4. Have good statistical randomness properties.

The basic efficiency criteria are discussed in the following subsections.

4.1 Linear Complexity

The linear complexity of a finite binary sequence S^n is the length of the shortest LFSR that generates a sequence having S^n and denoted by LC(S^n) and can be calculated by using Berlekamp-Massey Algorithm S^n [18].

4.2 The Periodicity

Let the P (S_i) represent the period of sequence S produce from the LFSR system, and let P(S_i) represent the period of each sequence produces from $LFSR_i$ for each $1 \leq i \leq N$. if $P(S_i) = 2^{r_i} - 1$; r_i is the lengths of $LFSR_i$, so the periodicity equal to:

$$P(S) = l.c.m\ (2^{r_1} - 1, 2^{r2} - 1, \dots, 2^{r_n} - 1),\ i = 1, \dots, N \quad (5)$$

Of course if $P(S_i)$ are relatively prime to each other ∀i, 1≤i≤k, then:

$$P(S) = \prod_{i=1}^{N} P(S_i) \quad (6)$$

The CF unit has no effect on the period of S generated from KG; rather, it is depends on the LFSR unit alone[11].

4.3 Correlation Immunity

Sequences that are merged by the Combining Function Unit (CF) $CF = F_N$ in the KG have a certain relationship to the sequence that CF produces as an output. This correlation occurs as a result of the highly nonlinear nature of the combined function, F_N. In general, the correlation probability (CP) of x represents the ratio of the number of comparable binaries (S_B) in two sequences to the length n of the compared portion of the two sequences.

$$CP = S_B/n \quad (7)$$

For $m = n - 1$ (where m is the total number of immune LFSRs), the optimal value of the immune correlation is all x_i ,$1 \leq i \leq n$ are independent on the value of Z in the output , and we considered it accepted if the value is in the range (45 to 55) [9].

4.4 Randomness

Assume $S = s_0, s_0, s_2, \dots, s_{n-1}$ is a binary set of length n. This section contains five tests that are commonly used to verify whether the sequences have the properties of a truly random sequence. It is underlined that the outcome of each test is not definitive, but rather probabilistic. Even if a sequence passes all five criteria, there is no certainty that it was generated by a random generator of bits [19].

1. **Frequency (Mono-bit) Test:** Its purpose is to determine whether the number of ones and zeros in (S) are nearly identical, as would be expected for a random bit-stream. Assuming n_0 indicate the number of zeros and n_1 indicate the number of ones in the sequence (S), the statistic employed is given by the equation (8).

$$x_1 = (n_0 - n_1)^2/n \quad (8)$$

Since n is equal to or higher than 10, almost follows a chi-squared allocation with 1 degree of freedom[20].

2. **Serial (Two-Bit) Test:** Its objective is to determine if the number of occurrences of 00, 01, 10, and 11 as S subsequences are nearly same, as predicted for a random bit-stream. Assuming $n0$ and $n1$ refer to the number of 0 and 1 in S, and $n00, n01, n10,$ and $n11$ refer to the number of 00, 01, 10, 11 in S, and considering that $n00 + n01 + n10 + n11 = n - 1$ because sub-sequences might overlap, the used statistic is equation (9):

$$x_2 = 4/n - 1(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - 2/n(n_0^2 + n_1^2) + 1 \quad (9)$$

When $n \geq 21$, almost chi-squared with 2 freedom levels [11].

3. **Poker Test:** If we assume that m is a positive number such that $[n/m] \geq 5, (2m)$, and we further assume that $k = [n/m]$, then we can divide the stream S into k non-overlapping parts, each of which is of length m, and assuming that (n_i) is the number of occurrences of the (i^{th}) sequence type of length m, $1 \leq i \leq 2m$, this test determines whether the streams of length m repeat almost the same number of times in S as would be suspected for a random sequence, the utilized statistic is equation (10).

$$x_3 = 2^m/k \left(\sum_{i=1}^{2m} n_i^2 \right) - k \quad (10)$$

Setting m = 1 in this testing results in the frequency test, almost exactly following a chi-squared distribution with $2^m - 1$ freedom degrees, as this test is a generic portion of the frequency test.

4. **Runs Test:** Its goal is to see if the sequence S contains the desired number of random runs (of 0's or 1's). $e_i = (n - i + 3)/2^{i+2}$ Predicts the number of i-length blocks in a random sequence of length n.

Using the assumption that k = the biggest integer i for which$e_i \geq 5$, (b_i) and (g_i) represent the number of blocks and gaps of length = i in S for each i $1 \leq i \leq k$,, the applicable statistic is given by the equation: (11).

$$x_4 = \sum_{i=1}^{k} \frac{(b_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(g_i - e_i)^2}{e_i} \quad (11)$$

Following a chi-squared distribution with $2k - 2$ freedom degrees and assuming gaps (g_i) looked at 0's distances and blocks (b_i) looked at 1's distances [21].

5. **Autocorrelation Test:** This operation seeks to identify correlations among streams S and (noncyclic) shifting versions. Assume d is a constant integer$1 \leq d \leq [n/2]$. The number of bits in $S \neq$ their d-shifts can be calculated as follows:

$A(d) \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}$, $\oplus$ represents the X-OR operator.

The utilized statistic is equation (12)

$$x_5 = 2(A(d) - \frac{(n-d/2)}{\sqrt{n-d}}) \quad (12)$$

Almost $N(0,1)$ for $n - d \geq 10$. For small $A(d)$ values that aren't intended to be large, use 2-sided a 2-sided must be utilized [21].

5. Design of New Stream Keygeneration

The encryption key is a vital part of any security system because the breach of the encryption key may lead to the collapse of the entire security infrastructure of the organization, so it is necessary to take care when designing the key. The data is encrypted and decrypted in our proposed system depending on a symmetric cryptosystem type, which uses the same key at the two communicating parties.

The proposed key generation part includes designing a new key generator which is hybrid between (**S**tream LFSR and **C**haotic maps)**,** that we named it as New Stream Chaotic Key Generator (SCKG). The proposed key generator design consists of the main three parts, these parts are described in details in the following subsections.

5.1 SCKG Components

The proposed **NSKG** includes the following main components:

1. **LFSR Unit (LFSRU):** Different prime lengths of the 8 LFSR's with different tapping polynomial functions:

   Tapping functions = [3 23; 7 29; 3 31; 2 37; 2 41; 1 43; 5 47; 3 53].
2. **Non-Linear Unit Function (NLUF):** This unit consists of 256 distinct and random polynomial bytes.
3. **5-Dimension Chaotic Maps (5DCM):** The used chaotic maps system is based on a 5 Dimensions Chaotic Maps which called 5DCM. The 5DCM were obtained in system (4).

5.2 Initialization of SCKG

At this stage of NSSCKG, we suggest using the random keys that we previously generated from the 5DCM to fill in the LFSRU, knowing that we have filled in the LFSRU only from the first two dimensions (x_i,y_i) of 5DCM.

The proposed filling method came to increase the complexity and the strength of the system, even the initial parameters of the generator are random keys. Also, in order to add another contribution, when we combine the use of LFSRU with chaotic system. The initialization is done as follows:

1. We have to obtain a string of bits from the first dimension(x_i) of the 5DCM to fill the LFSRU then complete the rest bits of LFSRU from the second dimension (y_i) of the 5DCM to get 296 bits ($k = 1,\dots,296$) because the length of all rejesters are 304 bits while the last cell of each regester are fill as pesented in the next point .
2. The last cells of each LFSR are filled by 1.
3. **LFSRU** moves to generate 256 distinct bytes to fill the **NLUF** with different and not repeated bytes. **Algorithm (1)** illustrates the Initialization of SCKG.

Algorithm (1) Initialization of SCKG:

Input: X,Y,Z,K,P
Output: initial bits for **LFSRU** and **NLUF**
Process: Step(1): From CM, by using (x_i,y_i) , obtain array IN(296) bits; Step(2): Filling the (8) LFSR's by the array IN; Step (3): The last cell of (8) LFSR's filled by 1's. Step(4): The **LFSRU** moves to fill the **NLUF** by 256 different and not repeated bytes; Step(5): STOP

5.3 Running of SCKG

Now, the **SCKG** is ready to move as explain below:

1. Moving the **LFSRU** to obtain an address bytes (*AD*) to **NLUF** s.t.:

$AD = \sum X_i * \frac{256}{2^i}$

Where X_i are output of LFSRi , $i = 1,2, \dots,8$.

2. The AD is input to **NLUF** to obtain the **S**tream **B**yte1 which called (**SBy1).**

SBy1= NLUF (AD)

3. Running chaotic maps one pulse to obtain **C**haotic **B**yte2 which called **(CBy2)** s.t:

CBy2 = Z1.

Z1 is the third dimention of 5DCM

4. Finally, we obtain the output key which we name as **S**tream **C**haotic **B**yte **K**ey (**SCBK**)s.t:

$SCBK = SBy1 \oplus CBy2$ (13)

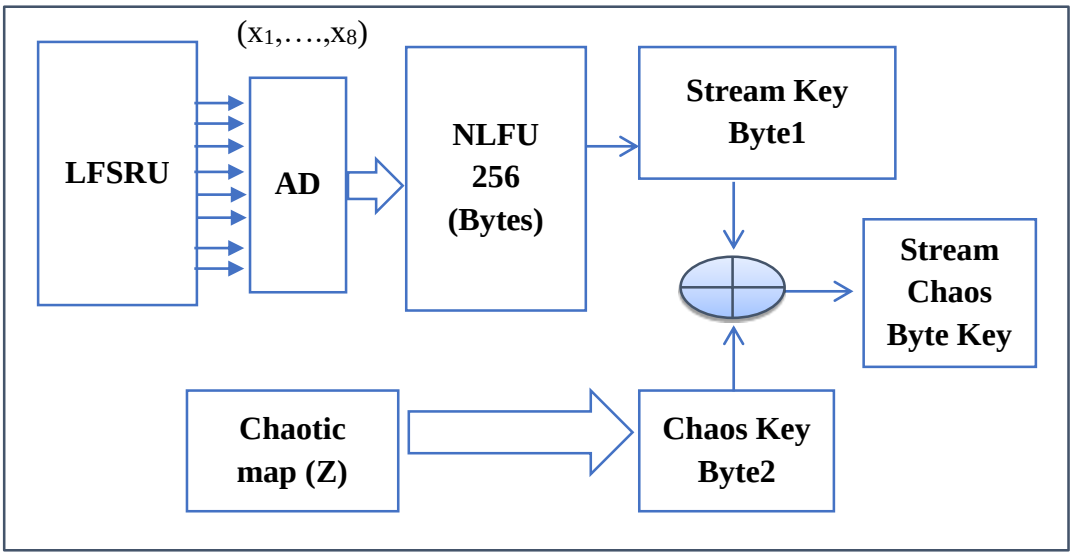

Figure 3. Block diagram of designing the SCKG.

6. Implementation of BEC on SCKG Output Key

In this section we will generate (3) random key files from the proposed SCKG to be tested by randomness and NIST tests and other tests. We will review the results of output keys from SCKG for various examples with different lengths (L_i, i=1, 2, 3), with different output key is as follows:

1. Example1: L1=512 KB (=4096bits).
2. Example2: L2=1024 KB (=8192 bits).
3. Example3: L3=2048 KB (=16192 bits).

6.1 Periodicity

In equation (14) we show the Periodicity of the sequence S which is generated from SCKG:

$$P(S) \approx 2^{23+29+31+37+41+43+47+52} = 2^{304} \qquad (14)$$

6.2 Linear Complexity

After applying perlikamp-Massey algorithm for the of the sequence S to find the linear complexity (LC) which is generated from the SCKG for the (3) example we obtain the results which are illustrated in **Table (1).**

In table (1 the liner complexity tests result for the three examples are shown.

Table 1. Linear complexity tests for the three examples.

Example1	Example2	Example3
2100	4301	8941

6.3 Correlation Immunity

Table (2) show the Correlation Immunity (CI) tests result for the three examples.

Table 2. Correlation Immunity tests result for three examples**.**

CI	CI decision		
	Example1	**Example2**	**Example3**
x_1	pass	Pass	Pass
x_2	Pass	Pass	Pass
x_3	Pass	Pass	Pass
x_4	Pass	Pass	Pass
x_5	Pass	Pass	Pass
x_6	Pass	Pass	Pass
x_7	Pass	Pass	Pass
x_8	Pass	Pass	Pass

Where x_j is the output of LFSR j of LFSRU, $1 \leq j \leq 8$.

CI (SCKG) =8 which equal the number of shift registers in running SCKG.

6.4 Key Space Test

Since as we depend on 5DCM which is a responsible for generating the permutation key, filling LFSRU and Z5D, so the 5 initial values of 5DCM be represent the key space for SCKG. If we take 16 digits accuracy for each real of $(x_0, y_0, z_{0,} p_0, k_0)$, so each initial key converted to 64 bits. So the key space (KS) can be calculated as follows:

$KS = 2^{5*64} = 2^{320}$.

Which mean from KS value that KG is ammune from Brut Forse attack.

7. NIST Tests on SCKG

After the key was successful in the previous tests, now we will be using NIST tests. Table (3) illustrate NIST tests for SCKG key.

Table 3. NIST tests for SCKG key.

Test No.	Test Name	Results
1	Rank	Agreed
2	Non-Periodic Templates	Agreed
3	Frequency	Agreed
4	Block Frequency	Agreed
5	Cumulative Sums	Agreed
6	Runs	Agreed
7	Longest Run	Agreed
8	Overlapping	Agreed
9	Universal	Agreed
10	Serial	Agreed
11	Lempel-Ziv	Failed
12	Linear Complexity	Agreed
13	Random Excursions Variant	Agreed
14	Random Excursions	Failed
15	Discrete Fourier Transform	Agreed
16	Approximation Entropy	Agreed

8. Conclusions and Future Works

1. The results of applying the basic efficiency criteria and NIST prove the efficiency of the proposed stream key generator SCKG.
2. From the key space test, we believe that the SCKG is immune against the brute force attack.
3. The proposed SCKG can be suitable to encrypt/decrypt all types of digital documents like, text, image, audio and video files.
4. More tests can be done on our key generator to ensure the efficiency of SCKG, like the CRYPT'X 98 tests.
5. Since that the proposed SCKG is immune against any attack, we call the military, civil and security institutes to benefit from this system to keep the classified documents are saved.

References

1. Ali, S.; Abdul, A. Data Security for Cloud Computing Based on Elliptic Curve Integrated Encryption Scheme (ECIES) and Modified Identity Based Cryptography (MIBC). *Int. J. Appl. Inf. Syst.* **2016**, *10*, 7–13, doi:10.5120/ijais2016451517.
2. Bhardwaj, I.; Kumar, A.; Bansal, M. A Review on Lightweight Cryptography Algorithms for Data Security and Authentication in IoTs. *4th IEEE Int. Conf. Signal Process. Comput. Control. ISPCC 2017* **2017**, *2017-Janua*, 504–509, doi:10.1109/ISPCC.2017.8269731.
3. Abdul-Hadi, A.M.; Saif-aldeen, Y. abdul-sahib; Tawfeeq, F.G. Performance Evaluation of Scalar Multiplication in Elliptic Curve Cryptography Implementation Using Different Multipliers Over Binary Field GF (2233). *J. Eng.* **2020**, *26*, 45–64, doi:10.31026/j.eng.2020.09.04.
4. Trappe, W. *Introduction to Cryptography with Coding Theory*; Pearson Education India, 2006; ISBN 8131762386.
5. Brandau, M.A. Implementation of a Real-Time Voice Encryption System. **2008**.
6. Luma, A.; Selimi, B.; Ameti, L. Using Elliptic Curve Encryption and Decryption for Securing Audio Messages. In Proceedings of the Transactions on Engineering Technologies: World Congress on Engineering 2014; Springer, **2015**,599–613.
7. Shree, D. Available Online at Www.Ijarcs.Info A Review on Cryptography , Attacks and Cyber Security. **2017**, *8*, 2015–2018.
8. Koblitz, N.; Menezes, A.; Vanstone, S. The State of Elliptic Curve Cryptography. *Des. codes Cryptogr.* **2000**, *19*, 173–193.
9. Ghazi, A.A.; Ali, F.H. Robust and Efficient Dynamic Stream Cipher Cryptosystem. *Iraqi J. Sci.* **2018**, *59*, 1105–1114, doi:10.24996/IJS.2018.59.2C.15.
10. Golomb, S.W. *Shift Register Sequences: Secure and Limited-Access Code Generators, Efficiency Code Generators, Prescribed Property Generators, Mathematical Models*; World Scientific, 2017; ISBN 9814632023.
11. Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. Handbook of Applied Cryptography. **2018**, doi:10.1201/9780429466335.
12. Naser, A.G.; Majeed, F.A.H. Constructing of Analysis Mathematical Model for Stream Cipher Cryptosystems **2017**, *58*, 707-715.

13. Sadkhan, S.B.; Mohammed, R.S. Proposed Random Unified Chaotic Map as PRBG for Voice Encryption in Wireless Communication. *Procedia Comput. Sci.* **2015**, *65*, 314–323, doi:10.1016/j.procs.2015.09.089.
14. Hobincu, R.; Datcu, O. A Novel Chaos Based PRNG Targeting Secret Communication. *2018 12th Int. Conf. Commun. COMM 2018 - Proc.* **2018**, *2018-January*, 459–462, doi:10.1109/ICComm.2018.8484795.
15. Naik, R.B.; Singh, U. A Review on Applications of Chaotic Maps in Pseudo-Random Number Generators and Encryption. *Ann. Data Sci.* **2022**, doi:10.1007/s40745-021-00364-7.
16. Kubba, Z.M.J.; Hoomod, H.K. Modified PRESENT Encryption Algorithm Based on New 5D Chaotic System. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *928*, doi:10.1088/1757-899X/928/3/032023.
17. Thesis, A.; In, B.; Fulfillment, P.; The, O.F.; For, R.; Abdul, F.; Hameed, R.; By, S. .86 **2017**.
18. Fúster-Sabater, A.; Cardell, S.D. Linear Complexity of Generalized Sequences by Comparison of PN-Sequences. *Rev. la Real Acad. Ciencias Exactas, Fis. y Nat. - Ser. A Mat.* **2020**, *114*, 1–18, doi:10.1007/s13398-020-00807-5.
19. Dołanaksoy, A.; Sulak, F.; Ułuz, M.; Şeker, O.; Akcengiz, Z. New Statistical Randomness Tests Based on Length of Runs. *Math. Probl. Eng.* **2015**, *2015*, doi:10.1155/2015/626408.
20. Elkamchouchi, H.M.; Saleh, G.A.; Saleh, Y.A. Efficient Speech-Based Random Number Generators. **2011**, *7*.
21. Alsaadi, A.A.; Naser Al-Shammari, A.G. Enhancement of Non-Linear Generators and Calculate the Randomness Test for Autocorrelation Property. *Iraqi J. Sci.* **2019**, *60*, 2229–2236, doi:10.24996/ijs.2019.60.10.17.