Enhancement of Non-Linear Generators to Calculate the Randomness Test for Frequency Property in the Stream Cipher Systems

Ibrahim Abdul Rasool Hammood¹, Ayad Ghazi Naser Alshamri²

¹Dept. of Mathematics, Collage of Science Directorate General for Vocational Baghdad University, Baghdad,
²Iraq Education Ministry, Baghdad, Iraq

https://doi.org/10.26782/jmcms.2019.10.00077

Abstract

In this paper, the key generators generated by using (Brüer generator, Geffe generator, and Linear generator), then improved these key generators (Briierand Geffe). In this research was the focus on the frequency test and then compares the outputs with results in a chi-square.

Keywords : Cryptography, Stream Cipher, Frequency, LFSR.

I. Introduction

The term cryptography has derived from a Greek term, in the current day context, it mentions the paraphernalia and Technologies used to make messages safe for telecommunication between the participants. Encryption is the study of information security sides such as confidentiality, data safety[I]. Encryption is the road to encrypt data. Basically, stream cipher was concentrated on bit based linear feedback shift registers LFSR. Stream ciphers are utilized for the Shared key cryptography. Previously the safeness and safety of data was handled by the computer, secret messages undergo different shapes of cryptography and cryptanalysis that, Cryptography is usually applied in practice to customize four services: confidentiality, authentication, data safety and prevention-Hackers. This flag has the largest place between sciences and is used in different areas including diplomatic, military, safeness and information; it also protects data [II].

Suggest a new stream cipher building principles on block cipher design. The basic rule is a replacement the building blocks used in block ciphers by equal stream cipher components.

In order to understand this process it works to build stream cipher very simple, that provides a lot of ease for hardware applications, it has a number of desirable encryption features.

Christophe De Canni`ere, "A Stream Cipher Construction Inspired by Block Cipher Design Principles", 2006, [III].
In previous years, a large number of encryption algorithms were proposed, but most faced problems such as lack of power and security, a new cryptographic algorithm has been suggested, a pseudo-random number is applied to control that the cryptographic method was chosen. "Chaotic encryption algorithm based on alternant of stream cipher and block cipher", 2010, Jianfeng Zhao, Zhenfeng Zhang, Xingyuan Wang, [IV].

Linear Feedback Shift Registers (LFSR) are applied as constructing blocks for much stream ciphers, where, an n-degree rudimentary contact polynomial is applied as a feedback function to recognize an n-bit LFSR.

This paper display that such LFSR are liable to strength analysis based Side Channel onslaught.

The main contribution of this sheet is the observation that the status of an n-bit LFSR can be specified by making gauges of power, 2007, "LFSR Based Stream Ciphers Are Vulnerable to Power Attacks", Debdeep Mukhopadhyay, Sanjay Burman, and Kamakoti Veezhinathan[V].

The suggested based encryption algorithms have proposed several new and effective ways to improve obtain image cryptographic techniques. This sheet presents effective randomness based on feedback stream cipher for image encryption system.

The bushy use of information-dependent reduplication, data-depend on inputs, and the inclusion of three independent feedback techniques. These suggested specifications are verified to supply a high-security grade; a complete specification for the suggested was presented. Key tests show that the proposed image encryption program shows a secure way to encrypt images at present and is transferred from encryption, Hamdy M. Kalash, Hossam El-din H. Ahmed, and Osama S. Farag Allah, 2005, "An Efficient Chaos-Based Feedback Stream Cipher (ECBFSC) for Image Encryption and Decryption"[VI].

II. Stream Ciphers

A stream cipher has a symmetrical cipher which cryptography the characters one after the other, where the stream key is converted over time, depending on the current inner status of the cipher, all stream ciphers are attempts to comprehend a cipher comparable to the One Time Pad (OTP) cipher but without its mistakes [VII].

Stream ciphers utilize a key and an initialization vector to construct a ‘pseudo-random’ stream key before the stream key is XOR together with the plaintext, it depends on how a stream cipher updates its inner status, stream ciphers can be classified into two groups: synchronous stream ciphers and self-synchronizing stream ciphers.

If the update functionality of the inner status is independent of the plain text or cipher text message, the cipher is categorized as a synchronous stream cipher.
On the opposed, in a case which the cipher develops the inner status depending on the former plain- or cipher text, the cipher is named a self-synchronizing stream cipher.

Usually, a stream cipher works at a much speed and has fewer hardware complications if compared with block ciphers [VIII].

### III. Linear Feedback Shift Register

A feedback shift register (FSR), it consists of two primary parts: a shift register and its feedback task. The SR is a chain of bits, (the tallness of an SR is calculated in bits). Each single time only one bit is wanted, every bit in the SR is transformed 1 bit to the right [VII].

Encryption experts have liked stream ciphers which consist of SR: Since they are easily applied in digital ciphering.

The feedback shift registers they are the short loop including digits of memory cell each that including one bit, the groups that cells from a register in every circle confirmed predefined groups of cells are tap and their value passed during a function named the feedback function, the register is then moved down through one bit with the make bit of the feedback shift register being the bit that is shifted outside the register.

The collection of the tapped bits is then fed into the flatulent cell at the upper of register and if feedback function is linear then it’s called Linear Feed a Shift Register and denoted by LFSR, a shift register from the length L on a Galois field $f$ consist of L register $[k_{L-1}, k_{L-2}, \ldots, k_0]$ with $k_r \in f_q$ and polynomial

$$p(x) = 1 + v_1 x + v_2 x^2 + \ldots + v_L x^L$$  \hspace{1cm} (1)

The including of $[k_{L-1}, k_{L-2}, \ldots, k_0]$ is named the case of LFSR and $P(x)$ named feedback or relation polynomial, the register is controlled by clock and every step the elements are shifted to the right so that $k = k + 1$ for $p = 0,1, \ldots, L - 2$ and $k_0$ is output the including of $k_{n-1}$ is calculation according to

$$k_{L-1} = v_1 k_{L-1} + v_2 k_{L-2} + \ldots + v_L k_0$$  \hspace{1cm} (2)

$R(x)$ is polynomial over $F_t$ which is rudimentary polynomial since it’s generate allelements of an extension fieldof $F_t$. An irreducible polynomial r(x) since (all primitive polynomial are irreducible polynomial) of degree n over $F_t$. P is prim number if the smallest positive integer m such that $r(x)$ divides $x^m - 1$ is m $p^n - 1$ if the feedback polynomial $P(x)$ is primitive polynomial of degree Leach of non-zero initial states produce an output sequence with maximum possible period $q^L - 1$ Figure (1).
Each LFSRs system contains a collection of linear shift registers. Each one shifted alone in one time as the nature of relation function and the combining function denoted by $CDF_n$ is a Boolean function on Galois field GF (2) it input are the sequence generated from each LFSR [8].

If $x_1, x_2, \ldots, x_n$ are input of $CDF_n$ such that $x_n \in GF(2); j=1, \ldots, n$ then:

$$CDF_n(x_1, x_2, \ldots, x_n) = a_0 + \sum_{j=1}^{n} a_j x_j + \sum_{i=1}^{n} \sum_{j=1}^{i} a_{ij} x_i x_j + \ldots + a_{1,2,\ldots,n} \sum_{j=1}^{n} x_j$$

... (3)

Where $a_0, a_1, a_{ij}, \ldots, a_{1,2,\ldots,n} \in GF(2)$ are the coefficients for combining of LFSR combined in function.

The block diagram of LFSR's System is shown in Figure(2).

**Figure (1)** the diagram of Linear Feedback Shift Register.

**Figure (2)** the block diagram of LFSR's System.

### IV. The Basic Efficiency Criteria (BEC) of Key Generator
#### IV.i. The Linear complexity

The linear complexity of a finite binary sequence $S^m$ is the length of the much shorter LFSR that creates a series as its initial condition and denoted by $LC(S^m)$ and can be calculated by using Berlekamp-Massey Algorithm [IX].

**Note**: it's important to note that the LFSR of minimallength that generates sequence $S^m$ of length $m$ is unique if and only if $m \geq 2L(S^m)$; where $L(S^m)$ is the linear complexity of sequence ($S^m$).

#### IV.ii. The Periodicity
Let the $\text{Per}(S)$ represent the period of sequence $S$ and let $\text{Per}(S_m)$ represent the period of each sequence produce from LFSR$_m$ for each $1 \leq m \leq n$; $k_m$ is the lengths of LFSR $m[X]$.

So the periodicity equal to:

$$\text{Per}(S) = \text{L.C.M} \left( 2^{k_m} - 1 \right); \quad 1 \leq m \leq n \quad (4)$$

**IV.iii. Correlation Immunity**

It's a connection between the series of $\text{CF} = F_n$ from the key generator and the sequences that are joint with every other by $\text{CF}$. This connection caused by the nonlinearity of the function $F_n$, the connection probability (CP) of $x$ in general, represents the ratio between the number of similar binaries ($N_Z$) of two series to the length $L$ of the compared part of them.

$$\text{CP} = \frac{N_Z}{L} \quad (5)$$

The correlation immune order can be computed from the logical truth table for combining function depending on computing $\text{CP}(x)$. The best correlation immune for any system when $m=n$; that's mean all $x_j$, $1 \leq j \leq n$ are independent of the output $Z$ and denoted $y \text{ CI} [XI]$.

**IV.iv. The Randomness**

This subsection presents five statistical tests that are usually used to select them whether the duple sequences possess some particular properties that are randomized series would be achievable to show.

It is confirmed another time that the outcome of every test isn't specified, but rather probabilistically. If a series succeeds in every five practical control tests, there isn't a warranty that it has actually created through a random bit generator.

It is essential to mention that the frequency, run, and autocorrelation tests are named the Primary random tests [IX].

To decide if the value of the test acquired is good sufficient for the sequence to pass, we have just to liken our value with a list of the Chi-square apportionment, with $\nu = k-1$ grade of freedom [VI]. Equations (6-10) represent the **Poker, Autocorrelation, Run, Frequency** and **Serial Test**.

i- **Poker test**

The poker test divides the sequence $S$ into $K$ parts with length $m \geq 3$, let $n_j$ be the observed number of occurrence of the $j^{th}$ type of a sequence of length $m$ this test locate whether

The sequence of length $m$ each show approximately the selfsame number of times in as would be foreseeable for a random sequence.

$$T_1 = \sum_{i=0}^{n} \frac{(n_i - E_i)^2}{E_i} \quad (6)$$

Where $E_i = C^n_j \left( \frac{1}{2^m} \right) \left( \frac{n}{m} \right)$; $T_1 \approx x^2(0.05, m)$. 

ii - **Autocorrelation test**
The autocorrelation test is used to check for correlations among the sequence S and (non-cyclic) shifted reversions of it. We define
\[ A(d) = \sum_{i=0}^{n-d-1} s_i + s_{i+d} \mod 2 \]
as the number of bits in the sequences that are equal to their d-shifts, the law used here [IV].

The run test is a subseries from S includes of sequential 0’s or 1’s, run of 0’s called Gap while a run of 1’s called Block.

The run test used to locate whether the number of runs of various lengths in the sequence S is as foreseeable for a random sequence [4].

\[ T_3 = \sum_{i=1}^{k} \left( \frac{(B_i - e_i)^2}{e_i} \right)^{1/2} \sum_{i=1}^{k} \left( \frac{(G_i - e_i)^2}{e_i} \right) \]

Where \( B_i \) and \( G_i \) are the number of blocks and gaps respectively of length \( i \) in S for each \( i, 1 \leq i \leq k \) the expected value equal to
\[ e_i = \frac{n-i+3}{2(i+2)}, T_3 \approx x^2(0.05, 2k - 2). \]

The frequency test use to define the digits from 0’s and 1’s in a series (key stream series) together length n are:

\[ T_4 = \sum_{j=0}^{1} \left( \frac{(n - \frac{n}{2})^2}{\frac{n}{2}} \right) = \left( \frac{n_0 - n_1}{n} \right)^2 \]

Where \( n_0, n_1 \) denoted the watched over digits with regard to 0’s and 1’s in S sequentially.
\[ \frac{n}{2} \] is a foreseeable amount of values, \( n \) equal to S in correlation to length \( S, T_4 \approx x^2(0.05, 1). \]

v - Serial Tests

This test aims to define if that the number of the appearance of 11, 01, 10, and 00 as a subsequence S are approximately the selfsame as would be predictable for random series

\[ T_5 = \sum_{j=0}^{1} \sum_{i=0}^{1} \left( \frac{n_{ij} - E}{E} \right)^2 \]

Where \( n_{00}, n_{10}, n_{01}, and n_{11} \) denoted the observed number of 00, 01, 10 and 11 respectively the foreseeable value \( E = \frac{(n-1)}{4} \), where \( n_{00} + n_{10} + n_{01} + n_{11} = n - 1 \);
\[ T_5 \approx x^2(0.05, 3). \]

V. Geffe generator[4]
Geffe generator in an example of nonlinear feedback shift registers system, linear feedback shift registers are insecure because they have a comparatively small linear complication and hence a comparatively small part of the key streams (LFSR sequence) can be applied to get the entire sequence by fix a set of linear equations. To increase the linear complexity of LFSR, one or more produce strings of LFSR's are joint with some nonlinear function to output comparatively high linear complication. Geffe generator is composed of three LFSRs of distinct lengths combined by a nonlinear function.

\[ f(z_1, z_2, z_3) = z_1z_2 + z_2z_3 + z_3 \]

VI. Brüer generator [5]

Brüergenerator in an example of nonlinear feedback shift registers system, linear feedback shift registers are insecure because they have a comparatively small linear complication and hence a comparatively small part of the key streams (LFSR sequence) can be applied to get the entire sequence by fix a set of linear equations. To increase the linear complexity of LFSR, one or more produce strings of LFSR's are joint with some nonlinear function to output comparatively high linear complication. Brüergenerator is composed of three LFSRs of distinct lengths combined by a nonlinear function.

\[ f(y_1, y_2, y_3) = y_1y_2 + y_2y_3 + y_3y_1 \]

VII. KeyManagement

Two types of keys considered as a primary key or LFSR of the system, this key are:

VII.i. Massage Key(MK):

This key is a code which is consists of (X) characters encrypted, new MK used with each new message to guarantee there no two messages that similar to this key is generated randomly before the encryption starter and it’s encrypted and delivers it with the encrypted message.

VII.ii. Basic Key (BK):

This key consists of (20) code character, this key shall be sent through a secure conduit.

VIII. Initialization

The grounds steps of initialization are:

1- Every character of BK transform to 8 bits then the series of BK has a length (20*8=160 bits).

\[ BK_1, ..., BK_{20} = BK_{1,1}, ..., BK_{4,5}, BK_{5,6}, ..., BK_{8,7}, ..., BK_{15,9}, ..., BK_{20,10}, \]  

Where \( BK_i \) is the BK character number \( i \); \( 1 \leq i \leq 20 \) and \( BK_{ij} \) is the bit \( j \) of \( BK_i \); \( 1 \leq j \leq 10 \).
2- In the same way each character of $MK$ transform to 8 bits then the series of $MK$ has a length (10*8=80 bits).
$MK_1, ..., MK_{10} = MK_{1,1}, ..., MK_{2,3}, MK_{3,4}, ..., MK_{5,6}, ..., MK_{10,10}$.
Where $MK_i$ is the $MK$ character number, $1 \leq i \leq 10$ $MK_{ij}$ is the bit $j$ of $MK_i$ $0 \leq j \leq 10$ [XII].

3- The series of elementary of system is the xor of $BK_{ij}$ and $MK_{ij}$ such that;
$MK_1, MK_2, ..., MK_{80}, MK_1, MK_2, ..., MK_{80}$
⊕
$BK_1, BK_2, ..., BK_{80}, BK_80, BK_{81}, BK_{82}, ..., BK_{160}$

\[ (11) \]

Where $I_j = M_j \text{xor} B_j$

4- From the series of LFSR’s are full of one by one and step by step as follows
A- LFSR 1 with length 41 bit (1:40).
B- LFSR 2 with length 51 bit (41:90).
C- LFSR 3 with length 71 bit (91:160).
And the last box of every LFSR full of with one.

5- Used the keys generators Geffe, Brüer, and Liner generator.

6- Add an LFSR4 to the key generator Geffe and the key generator Brüer.

7- Used a frequency test to check the passage or failure of all the keys generated.
The algorithm of the suggested system for the frequency test

\begin{itemize}
  \item **Step (1):** input $BK$ (20 letters), $MK$ (10 letters), length $L$ bytes.
  \item **Step (2):** used the binding function $\text{xor}$ between $BK$ and $MK$.
  \item **Step (3):** initialization of the basic shift register system.
  \item **Step (4):** used lengths of LFSR 1 with length (41 bit) and LFSR 2 with length (51 bit) and LFSR 3 with length (71 bit).
  \item **Step (5):** the last box of every LFSR full of with one.
  \item **Step (6):** used the keys generators Geffe, Brüer, and Liner generator.
  \item **Step (7):** results improved when adding an LFSR4 to the key generator Geffe and the key generator Brüer.
  \item **Step (8):** used one of the Randomness Tests (frequency test) to check the passage or failure of all the keys generated.
  \item **Step (9):** output results of the frequency test.
\end{itemize}

END

The diagram of the algorithm is described in **Figure (3).**
Example:

In this example, keys generator was used with different lengths 5000, 10000 and 50000 depending on Form the different of BK and MK it was used in this Example BK(20character)and it was used MK (10character).

Table (1) the Frequency test on linear generator

| length  | number of one | number of zero | The result of Test | difference between '1'&'0' | Decision |
|---------|---------------|---------------|--------------------|--------------------------|----------|
| 5000    | 2557          | 2443          | 2.599              | 114                      | Success  |
| 10000   | 4933          | 5067          | 1.796              | 134                      | Success  |

Figure (3) the diagram of the algorithm
In table (1) the linear keys generator was used by using three different lengths of keys.

**Table (2)** the Frequency test on Geffe generator and enhanced Geffe generator.

| generator | length | number of one | number of zero | The result of Test | difference between '1' & '0' | Decision |
|-----------|--------|---------------|----------------|---------------------|----------------------------|----------|
| Geffe     | 5000   | 2401          | 2599           | 7.841               | 198                        | failure  |
|           | 10000  | 4946          | 5054           | 1.166               | 108                        | Success  |
|           | 50000  | 25041         | 24959          | 0.134               | 82                         | Success  |
| improveGeffe | 5000   | 2463          | 2537           | 1.095               | 74                         | Success  |
|           | 10000  | 4962          | 5038           | 0.578               | 76                         | Success  |
|           | 50000  | 24904         | 25096          | 0.737               | 192                        | Success  |

In table (2) Geffe keys generator was used and this generator has been improved and it was using three different lengths of keys.

**Table (3)** the Frequency test on Brüer generator and enhanced Brüer generator.

| generator | length | number of one | number of zero | The result of Test | difference between '1' & '0' | Decision |
|-----------|--------|---------------|----------------|---------------------|----------------------------|----------|
| Brüer     | 5000   | 2426          | 2574           | 4.381               | 148                        | failure  |
|           | 10000  | 4933          | 5067           | 1.796               | 134                        | Success  |
|           | 50000  | 24979         | 25021          | 0.035               | 42                         | Success  |
| improveBrüer | 5000   | 2513          | 2487           | 0.135               | 26                         | Success  |
|           | 10000  | 5013          | 4987           | 0.068               | 26                         | Success  |
|           | 50000  | 24893         | 25107          | 0.916               | 214                        | Success  |

In table (3) Brüer keys generator was used and this generator has been improved and it was using three different lengths of keys.

9. Conclusions:

The deduced that could be generated from this work are as follows:
1. Two types of keys generators results have been improved (Geffe and Brüer) for better results.

2. The result depends on BK and MK and when BK and MK are changed, the result change too.

3. The observation that the greater the length of the key the greater is the rate to pass the key generated through frequency test.

4. It has been noticed that the best results obtained are the keys generated by the linear key generator.

References:

I. C. Paar, J. Pelzl, 2010, "Understanding Cryptography", Springer, Verlag Berlin Heidelberg.

II. A. Klein, 2013, "Stream Ciphers", Springer Verlag London.

III. Fardous Eljadi, 2017, "Dynamic Linear Feedback Shift Registers: A Review", Kuala Lumpur, Malaysia.

IV. Alice Reinaudo, 2015, "Empirical testing of pseudo random number generators based on elliptic curves", Linnaeus University, Sweden.

V. Yassir Nawaz, 2007, "Design of Stream Ciphers and Cryptographic Properties of Nonlinear Functions", University of Waterloo, Canada.

VI. Hossam El-din H. Ahmed, Hamdy M. Kalash, and Osama S. Farag Allah, 2005, "An Efficient Chaos-Based Feedback Stream Cipher (ECBFSC) for Image Encryption and Decryption", Menoufia University.

VII. M. Robshaw and O. Billet (Eds.), 2008, "Design of a New Stream Cipher—LEX", Springer-Verlag Berlin Heidelberg.

VIII. Roy Ward, Tim Molteno, 2007, "Table of Linear Feedback Shift Registers", University of Otago.

IX. Abdullah Ayad Ghazi, Faez Hassan Ali, 2018, "Robust and Efficient Dynamic Stream Cipher Cryptosystem", Iraqi Journal of Science, University of Baghdad.

X. Xingyuan Wang, Xiaojuan Wang, Jianfeng Zhao, Zhenfeng Zhang, 2010, "Chaotic encryption algorithm based on alternant of stream cipher and block cipher", Springer, Dalian University of Technology.
XI. Gordon Meiser, 2007, "Efficient Implementation of Stream Ciphers on Embedded Processors", Ruhr-University Bochum (Germany).

XII. Rusol M. Shaker Alzewary, Ayad G. Naser Al-Shammar, 2016, "Design of High Efficiency Non-linear Keys Generator Based on Shift Registers", Iraqi Journal of Science, University of Baghdad.