Using the Non-Linear Generator to Calculate the Randomness Test for Frequency Property And use it to encrypt and decrypt message by using the Matlab program

Ibrahim Abdul Rasool Hamoud¹, Ayad Ghazi Naser Alshamri²

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

Email: ¹Ibrahimalkaby2000@gmail.com, ²dr_ayad64@vocational.edu.iq
dr_ayad64@yahoo.com

http://doi.org/10.26782/jmcms.2019.10.00024

Abstract

In this paper, some of the key types used in the encryption system are displayed, and one type of key generator is displayed (Geffe generator). Matlab 2017 also uses some interfaces to illustrate the frequency test on the Encryption keys. Also, interfaces are displayed for encrypting and decrypting a message.

Keywords: Encryption, Frequency, LFSR, Decryption

I. Introduction

Several years ago, encryption became an important and essential science for several years; Encryption is the science of Provides safe services. There are two keys, encryption of the secret key and encryption of the general key, thoughts to do cryptography to the secret key system. Encryption is a type of secret technology used by special entities or the government. Today encryption is at the core of many safe applications Such as banking, government services, cell phones, military, and security [1].

That offers a process of safe scalable flowing (SSS) that can less-complication and large quality Code to be completed at middle, possibly untrusted, network nodes without compromising the end-to-end security of the system. (SSS) Video encryption into safe scalable packets use jointly intended scalable coding and advanced encryption technicality. This group let downstream transcoders to implement transcoding operations such as bitrate lowering and spatial down sampling by simply cut or discard packets, and With no Encryption the data, John Apostolopoulos, S.J. Wee,” Secure scalable streaming enabling transcoding without decryption”,2001, [IV].

Digital images have been connected dramatically online. The safety of digital images Is a basic and important rule on the shared telecommunication canal. Different
technicality is applied to protect the digital image, such as encryption, Hide information and tags these are the methods for the safety of digital images, “Image encryption using block based transformation Algorithm”, 2019 [VII].

The important rule is to convert, pixel by pixel, Color the image into logistic maps with a one-way mess along with the initial conditions. After very few number repetitions and cycles, the image becomes unable to distinguish because of the characteristics of chaotic systems. Since the maps are coupled, It is possible to return the entire image using the decoding algorithm if the map parameters are known, the number of duplicates is known, the number of cycles is known, and the image size is known, A. N. Pisarchik, M. Carpio-Valadez and N. J. Flores-Carmona,2006 [V].

Stream Encryption is part of the encryption system in which normal text is divided into little parts, the letter is called and a single character is encoded, at the time unencrypted text is encrypted little by little. Stream Cipher systems, in stream ciphers, Message modules are bit units, a key is usually created by a random bit generator (see Figure (1)). The plaintext is encrypted on a bit-by-bit basis.

![Stream Cipher System](image)

The key is fed into a random bit generator to create a long sequence of binary signals. This “key stream” k is then mixed with plaintext m, usually by a bitwise XOR (Exclusive-OR modulo 2 addition) to produce the ciphertext stream, using the same random bit generator and seed, the security of stream cipher is thus always measured relative to the complexity of exhaustive Searching for the correct key. If the complexity of an attack is less than that of the exhaustive search, the cipher is said to be broken [VI].

In military encryption, we can create a cryptographic flow in a separate box that is subject to strong security rules and is fed on devices such as a variety of radios that perform an XOR portion of its operation. The final device can then be designed for use in a less robust and rigid environment [X].

Encryption consists of two parts; the first is the encryption of messages and it is the same system of confidentiality. The second cryptanalysis is concerned with
decoding; today we need a very cryptographic system because of the need for encryption in various fields, which is a very important science [III].

II. Generate encryption keys for the system

Before explaining the basic steps of an open cryptographic system, it is very important to display some specifics in the encryption keys and signal to one of the types of key generators [XIII].

III. Key Management

It is possible for the proposed system to operate in two main processes when a center has several stations [I].

III.i. General Information Key (KIG) [XI]:

This contains general information sent by the Center to stations connected to the mode of transport, which means that the encryption key is only used by the Center and the stations are used to decrypt, such that KIG1=KIG2=...=KIGm.

As usual, information sent in one direction by instructions is represented, orders,…, etc.

III.ii. Special Information Key (KIS) [IX]:

This contains transfer the data between the center and every station attached to the center in two ways transfer. The message Sent from the center and station z is encrypted in key various from the key used to encrypt message from station z to the center, 1≤z≤m and theKIS_{z}^{s} ≠ KIS_{z}^{r} (send (s) or receive (r)). Figure (2) shows the key management for m stations; notice that the number of keys for KIG is m, while for KIS is 2m.

![Fig.2.Key Management for m stations.](image-url)
In order to fill the system with the seed key, we propose using two kinds of keys that are designed as the first key for LFSRs, these keys are:

IV. Massage Key (MK):

MK is a key that consists of ten encrypted characters and uses a new MK part with each message to make sure that there are no messages that are similar to this key randomly before encrypted and it is delivered with the encryption message [1].

V. Basic Key (BK):

This key consists of (20) code character, this key shall be sent through a secure conduit [12].

VI. Geffe generator [V]

Geffe generator at an epitome of nonlinear feedback shift registers system, linear feedback shift registers are insecure because they have a comparatively little linear complexity and hence a relatively little portion of the key streams (LFSR sequence) can be used to obtain the whole series by limitation a group of linear equations. To altitude, the linear complexity of LFSR, one or more output strings of LFSR's are joint with some nonlinear function to output relatively high linear complication. Geffe generator is Consists of three LFSRs of different lengths combined by a nonlinear function

\[ f(k_1, k_2, k_3) = k_1k_2 + k_2k_3 + k_3 \]

VII. Bank of LFSR's [X]

that bank means an information folder and contains 3 LFSR that will be various and will be multilevel for connection and different lengths, subsequently the wanted numbers of LFSR for MS and BS systems that we equipping to the suggested main generator and then stock in a particular memory and safe by confidential key which is portion From the main key

VIII. Initialization

The grounds steps of initialization are:

1- Each letter of BK convert to 8 bits thereafter the chain of BK has a length (25*8=200 bits).

\[ BK_1, BK_2, ..., BK_{25} = BK_{1,1}, ..., BK_{4,5}, BK_{5,6}, ..., BK_{8,7}, ..., BK_{15,9}, ..., BK_{25,15} \], Where \( BK_i \) is the BK character number; \( 1 \leq i \leq 25 \) and \( BK_{ij} \) is the bit \( j \) of \( BK_i \), \( 1 \leq j \leq 15 \).

2- In the same road every letter of MK convert to 8 bits then the chain of MK has a length (15*8=120 bits).

\[ MK_1, ..., MK_{15} = MK_{1,1}, ..., MK_{2,3}, MK_{3,4}, ..., MK_{7,8}, ..., MK_{15,15} \].

Where \( MK_i \) is the MK character number, \( 1 \leq i \leq 15 \), \( MK_{ij} \) is the bit \( j \) of \( MK_i \), \( 0 \leq j \leq 15 \) [IV].

3- The series of elementary of system is the \textbf{xor} of \( BK_{ij} \) and \( MK_{ij} \) such that;
\( MK_1, MK_2, ..., MK_{120}, BK_1, BK_2, ..., BK_{200} \)

\[ \oplus \]

\( BK_1, BK_2, ..., BK_{120}, BK_{121}, BK_{122}, ..., BK_{200} \)

\[ I_1, I_2, ..., I_{120}, I_{121}, I_{122}, ..., I_{200} \]

\[ \text{(1)} \]

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 53 bit (1:52).
B- LFSR 2 with length 73 bit (53:102).
C- LFSR 3 with length 91 bit (102:200).

And the last box of every LFSR full of with one.

5- Applied the keys generators Geffe.

6- insert an LFSR 4 to the key generator Geffe.

7- It is applied the frequency test until the crossing or the failure of all the keys have been verified. Proposed system algorithm for frequency testing [XII]

Step (1): input BK (25 letters), MK (15 letters), length L bytes.

Step (2): applied the binding function xor between BK and MK.

Step (3): initialization of the base shift register system.

Step (4): the applied lengths of LFSR 1 with length (53 bit) and LFSR 2 with length (73 bit) and LFSR 3 with length (91 bit).

Step (5): the final box of each LFSR full of with one.

Step (6): applied the keys generators Geffe.

Step (7): outcomes have been improved when adding an LFSR4 to the key generator Geffe.

Step (8): Applied one random test (frequency test) to verify the crossing or failure of all keys created

Step (9): product results of the frequency test.

END
IX. Configuration for the encryption system

Some of the interfaces used by Matlab 2017 will be displayed in order to display the key types used in the program. The frequency test results will be displayed on the keys used in this search. Also, some of the interfaces used in text encryption are displayed and then another interface is displayed showing decryption of this text [VIII].

These interfaces are displayed in the following:

A. The Frequency test for the output keys

The GUIDE of Frequency test (FT) program for all output key test is described in the block diagram for (FT) output key test.

![fig.3](image)

**fig.3.** the block diagram for (FT) output key test.

B. The security interface of the user program

This interface is used to protect the program from intruders is described in Figure (4).

![Fig.4](image)

**Fig.4.** the diagram for protected the program.
C. The Encryption and Decryption process using ENLG

In the format, the main interface used for text encryption and decryption is displayed, is described in Figure (5).

![Fig.5. the main window of ENLG program.](image)

D. The Encryption of text by using ENLG

In the format, The message is encrypted by entering the MK and BK key length as well as inserting the message to be encrypted, is described in Figure (6).

![Fig.6. the Encryption window of ENLG program.](image)

E. The Decryption of text by using ENLG

In the format, a message is decrypted, is described in Figure (7).
Fig. 7. The Decryption window of ENLG program.

Example 1

Used the Geffe generator to generate keys with different lengths 5000, 10000 and 50000 depending on different BK and MK we take in this example BK (25 characters) and we take MK (15 characters), The frequency test will be performed on these keys. The results will be displayed in the table (1).

Table (1) the Frequency test on Geffe generator

| Generator | Length | Number of one | Number of zero | The result of Test | Difference between '1' & '0' | Decision |
|-----------|--------|---------------|----------------|-------------------|-----------------------------|----------|
|           | 5000   | 2401          | 2599           | 7.841             | 198                         | fail     |
| Geffe     | 10000  | 4946          | 5054           | 1.166             | 108                         | pass     |
|           | 50000  | 25041         | 24959          | 0.134             | 82                          | pass     |
Example 2

Used the Geffe generator to generate keys with different lengths 10000, 100000 and 500000 depending on different BK and MK we take in this Example BK(25 characters) and we take MK (15 characters). The frequency test will be performed on these keys. The results will be displayed in the table (2).

Table (2) the Frequency test on Geffe generator

| Generator | Length | Number of one | Number of zero | The result of Test | Difference between '1'&'0' | Decision |
|-----------|--------|---------------|----------------|--------------------|---------------------------|----------|
| Geffe     | 10000  | 5036          | 4964           | 0.518              | 72                        | pass     |
|           | 100000 | 50149         | 49851          | 1.505              | 298                       | pass     |
|           | 500000 | 250392        | 249608         | 1.229              | 784                       | pass     |
Example 3

Used the Geffe generator to generate keys with different lengths 100000, 500000 and 1000000 depending on different BK and MK we take in this Example BK(25 characters) and we take MK (15 characters), The frequency test will be performed on these keys. The results will be displayed in the table (3).

Table (3) the Frequency test on Geffe generator

| Generator | Length  | Number of one | Number of zero | The result of Test | Difference between '1' & '0' | Decision |
|-----------|---------|---------------|----------------|--------------------|----------------------------|----------|
| Geffe     | 100000  | 50019         | 49981          | 0.014              | 38                         | pass     |
|           | 500000  | 250339        | 249661         | 0.919              | 678                        | pass     |
|           | 1000000 | 500757        | 499243         | 2.292              | 1514                       | pass     |

X. Conclusions:

The deduced that could be generated from this work are as follows:

1. One kind of keys generator outcomes have been improved (Geffe) for best outcomes.

2. The outcomes rely on BK and MK and when BK and MK are changed, the outcome changes also.

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. To increase security protection, you can refer to a secure encryption system and then encrypt the ENLG master files.

5. Encryption system can not only be used in text encryption; it can be used in many fields including audio and video encryption because it is very secure and very fast.

6. In the safeguarding process, we make sure that ENLG is protected from any offensive and can be run on any computer.
References:

I. D. Watanabe, S. Furuya, H. Yoshida, and B. Preneel, "A New Keystream Generator MUGI", 2002, University of Leuven, Belgium. *Modular Addition*, 2006, University of Leuven, Belgium.

II. Fischer, S, 2008 “Analysis of Lightweight Stream Ciphers” M Sc. thesis. Department of Physics, University of Berne of nationality Suisse.

III. Ibrahim Abdul Rasool Hamoud, Ayad Ghazi Naser, 2019, "Enhancement of Non-Linear Generators to Calculate the Randomness Test for Frequency Property in the Stream Cipher Systems", Iraqi Journal of Science, University of Baghdad.

IV. John Apostolopoulos, S.J. Wee, 2001, “Secure scalable streaming enabling transcoding Without decryption ”, Thessaloniki, Greece.

V. Kevin M. Cuomo and Alan V. Oppenheim, 1993, "Circuit implementation of synchronized Chaos with applications to communications", Massachusetts Institute of Technology, Cambridge.

VI. Rusol M. Shaker Alzewary, Ayad G. Naser Al-Shammar, 2016, "Design of High Efficiency"

VII. Sam Higginbottom, 2019, “Image encryption using block based transformation Algorithm ”, University of Agriculture Technology and Sciences, India.

VIII. Souradyuti Paul, Bart Preneel, “On the (In) security of Stream Ciphers Based on Arrays and Non-linear Keys Generator Based on Shift Registers”, Iraqi Journal of Science, University of Baghdad.

IX. Lawrence C. Washington, 2008, “Elliptic Curves Number Theory and Cryptography”.

X. Mattsson, J., 2006 “Stream Cipher Design”, M S thesis Department of Computer Science, at The School of Engineering Physics, University of Royal Institute of Technology, Stockholm, Sweden.

XI. Neal Koblitz, “An elliptic curve implementation of the finite field digital signature Algorithm”, University of Washington.

XII. Thomas Peyrin, 2016, “Fast Software Encryption”, University of Berlin, Germany.

XIII. Yassir Nawaz,”Design of Stream Ciphers and Cryptographic Properties of Nonlinear Functions”, Waterloo, Ontario, Canada, 2007.