ISSN: 0067-2904

Constructing of Analysis Mathematical Model for Stream Cipher Cryptosystems

Ayad Ghazi Naser, Fatin A. H. Majeed*

Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq.

Abstract

The aim of this paper is to construct the analysis mathematical model for stream cipher cryptosystems in order to be cryptanalysis using the cryptanalysis tools based on plaintext attack (or part from it) or ciphertext only attack, choosing Brüer generator as study case of nonlinear stream cipher system.

The constructing process includes constructing the linear (or non-linear) equations system of the attacked nonlinear generator. The attacking of stream cipher cryptosystem means solving the equations system and that means finding the initial key values for each combined LFSR.

Keywords: Cryptology, Cryptanalysis, LFSR, Stream Cipher, Ciphertext only attack, Brüer Generator

بناء نموذج تحليلي رياضي لنظم التشفير الانسيابي

اياد غازي ناصر، فاتن حميد مجيد*

قسم الرياضيات، كلية العلوم، جامعة بغداد، بغداد، العراق.

الخلاصة

هذا البحث يهدف الى بناء نموذج تحليلي رياضي لنظم التشفير الانسيابي لاستخدام ادوات التحليل على اسلوب المهاجمة باستخدام النص الواضح (او جزء منه) او اسلوب المهاجمة بتوفر النص المشفر باستخدام مولد برور كحالة دراسية لنظم التشفير الانسيابي.

ان عملية البناء تتضمن بناء نظام معادلات خطية (او لا خطية) للمولدات المراد مهاجمتها. ان مهاجمة نظم التشفير الانسيابي يعني حل نظام المعادلات الخطية وهذا بدوره يعني ايجاد القيم الابتدائية للمسجلات الزاحفة المركبة في المولد.

1. Introduction

The goal of **Cryptography** is to build systems that are hard to identify and **Cryptanalysis** is the science and study of methods of breaking ciphers. It is a system identification problem [1]. To attack a cryptosystems successfully the cryptanalysis is forced to be based on subtle approaches, such as knowledge of at least part of the plaintext encrypted, knowledge of characteristic features of the used language,..., with some luck. However, in practice, some of this information may be inaccurate, imprecise, or missing, which, in turn, causes to decrease the possibility of attacking and increasing the time or the resources required by the analyst. The **Cryptosystem** are the systems which use the encryption and decryption processes.

There are essentially two different types of cryptosystems, these cryptosystems are: **public key** and **secret key** cryptosystems [2]. First let us defined some important notations:

- **P** is the Plaintext message and **C** is the Ciphertext message.

*Email: tonaabd90@gmail.com

- **Key space K**: a set of strings (keys) over some alphabet, which includes the encryption key e_k and the decryption key d_k.
- The **Encryption** process (algorithm) E: $Ee_k(P) = C$.
- The **Decryption** process (algorithm) D: $Dd_k(C) = P$.
- The algorithms E and D must have the property that: $Dd_k(C)=Dd_k(Ee_k(P))=P$.

The public key cryptosystem also called **asymmetric cryptosystems**. In a public key (**non-secret key**) cryptosystem, the encryption key e_k and decryption key d_k are different, that is $e_k \neq d_k$.

The secret Key Cryptosystem also called **symmetric cryptosystems**. In a conventional secret-key cryptosystem, the same key ($e_k=d_k=k \in K$), called **secret key**, used in both encryption and decryption; we are interest in this type of cryptosystems.

There are many different types of secret key cryptosystems, like monographic (character) ciphers, polygraphic (block) ciphers, exponentiation ciphers and stream (bit) ciphers in which we shall focus [3].

Juntao G. and et al in their paper in 2006 [4], propose a fault attack on the Balanced Shrinking Generator as one of an important symbol of stream cipher. The results show that the attacker can obtain the secret key by analyzing faulty output sequences which is produced by changing control clock of one of LFSR.

Ali in 2007 [5], in his paper propose a cryptanalysis attack algorithms, called GA-Cryptanalysis system (GACS), on stream cipher systems using plaintext attack, choosing three cases study as symbols of stream cipher in the performance of GA.

The rest of this paper is organized as follows: in section 2 we discuss the stream cipher concept; in section 3 the study cases and some types of attacking of this paper are discussed. The proposed cryptanalysis system for stream cipher cryptosystems of this paper are detailed in section 4.

2. Stream Cipher systems

In **stream ciphers**, the message units are bits, and the key is usual produced by a **random bit generator** (see Figure -1)). The plaintext is encrypted on a bit-by-bit basis.

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. Shift register sequences are used in both cryptography and coding theory. There is a wealth of theory about them; stream ciphers based on shift registers have been the workhorse of military cryptography since the beginnings of electronics.

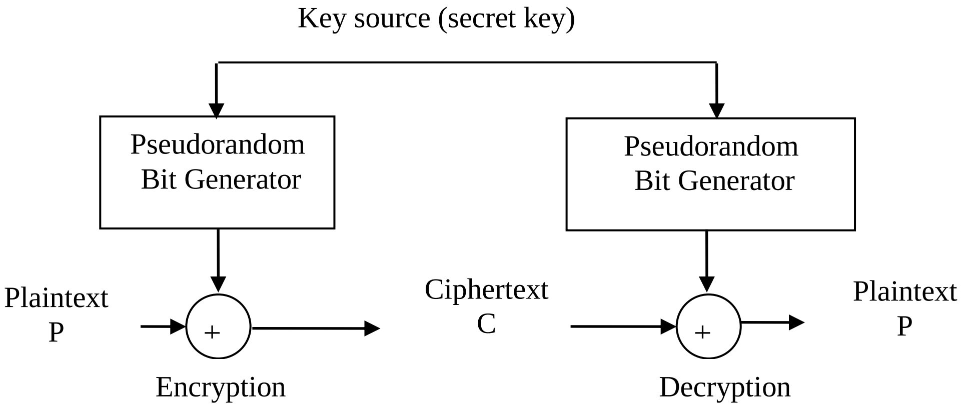

Figure1-stream cipher system.

The key is fed into random bit generator to create a long sequence of binary signals. This "key-stream" k is then mixed with plaintext m, usually by a bit wise XOR (Exclusive-OR modulo 2 addition) to produce the ciphertext stream, using the same random bit generator and seed.

Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry. They are also more appropriate, and in some cases mandatory (e.g., in some telecommunications applications), when buffering is limited or when characters must be individually processed as they are received. Because they have limited or no error propagation, stream ciphers may also be advantageous in situations where transmission errors are highly probable [6].

Linear Feedback Shift Register (LFSR) systems are used widely in stream cipher systems field. A LFSR System consists of two main basic units. First, is a LFSR function and initial state values [3]. The second one is, the Combining Function (CF), which is a boolean function [7]. Most of all stream cipher systems are depend on these two basic units.

3. Attacking the Stream Cipher Cryptosystem

To attack the stream cipher system, two types of attacks will be adopted. First, the plaintext (probable word) attack, and the second is the cipher only attack. Before we discuss these types of attacks, we pick two study cases to be attacked using the adopted attacks.

3.1 Study Cases

3.1.1 Single LFSR Cryptosystem

The first case study which we want to attack, is a single LFSR, which this system has no combining function, so we expect that it's can be expressed as one linear equations system.

Most practical stream-cipher designs center around LFSR. In the early days of electronics, they were very easy to build. A shift register is nothing more than an array of bit memories and the feedback sequence is just a series of XOR gates. A LFSR-based stream cipher can give you a lot of security with only a few logic gates.

3.1.2 Brüer Cryptosystem

The combining function of this cryptosystem called Threshold function, and known as majority function too. It’s called so, since this cryptosystem consists of odd number of LFSR’s, so naturally, there are a majority in the output bits for one from another, this mean, which one be the major, it will be the output. So it can be represented by following equation [8]:

$$z = \begin{cases} 1, & \text{when } \sum_{i=1}^{n} x_i > \frac{n}{2} \\ 0, & \text{when } \sum_{i=1}^{n} x_i < \frac{n}{2} \end{cases} \qquad \dots \qquad (1)$$

where

n : is positive integer odd number.

x_i: output of LFSR i.

z: final output.

We expect that we can express each LFSR by one linear equations system, but, each has unknown absolute values, since the output of each LFSR is unknown. Then, we concatenate the constructed linear equations system with each other using the nonlinear combining function to construct one nonlinear equations system with known absolute values, since the output sequence of Brüer cryptosystem is known.

3.2 Known Plaintext (Probable word) Attack

In this type of attack we assume that we have exact word (part of plaintext) is known, that means we have actual output key obtained from the stream cipher cryptosystem. This kind of attack allows to apply two techniques, the first represented by constructing a Linear Equations System (LES) with known absolute values (actual key); while the second option make us to change the analysis by estimating the initial values of combined LFSR's, then specify the actual one by compared the corresponding output key with actual key. In the following subsection we will discuss the two techniques.

Before involving in solving the LES or Nonlinear Equations System (NES), it should show how could be the LES for a single LFSR or NES for Brüer cryptosystem constructed. Let’s assume that all LFSR that are used are maximum LFSR (m-LFSR), that means, Period $P=2^L-1$, where L is LFSR length.

3.2.1 LES for Single LFSR

Let SR_L be a just single LFSR with length L, let $A_0=(a_{-1},a_{-2},\ldots,a_{-L})$ be the initial value vector of SR_L, s.t. a_{-j}, $1\le j\le L$, be the component j of the vector A_0, in another word, a_{-j} is the initial bit of stage j of SR_L, let $C_0^T=(c_1,\ldots,c_L)$ be the feedback vector, $c_j\in\{0,1\}$, if $c_j=1$ this means that the stage j is connected else it's not. Let $S=\{s_i\}_{i=0}^{r-1}$ be the sequence (or $S=(s_0,s_1,\ldots,s_{r-1})$ read “S vector”) with length r generated from SR_L. The generation of S depending on the following equation:

$s_i = a_i = \sum_{j=1}^{L} a_{i-j} c_j \quad i=0,1,\ldots, r\text{-}1$ (2)

Equation (2) represents the linear recurrence relation [9].

The objective is finding the A_0, when L, C_0 and S are known.
Let M be a L×L matrix, which is describes the initial phase of SR_L
$M=(C_0| I_{L\times L-1})$, where $M^0=I$.
Let A_1 represents the new initial of SR_L after one shift, s.t.

$$A_1=A_0\times M=(a_{-1},a_{-2},\ldots,a_{-L})\begin{pmatrix} c_1 & 1 & \cdots & 0 \\ c_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ c_L & 0 & \cdots & 0 \end{pmatrix} = (\sum_{j=1}^{L} a_{-j} c_j, a_{-1},\ldots,a_{1-L}).$$

In general,
$A_i=A_{i-1}\times M, \quad i=0, 1, 2, \ldots, r\text{-}1$ (3)
Equation (3) can be considered as a recurrence relation, so we have:
$A_i=A_{i-1}\times M=A_{i-2}\times M^2=\ldots=A_0\times M^i$ … (4)

The matrix M^i represents the i phase of SR_L, equations (3) and (4) can be considered as a Markov Process s.t., A_0, is the initial probability distribution, A_i represents probability distribution and M be the transition matrix [7].
notice that:
$M^2=[C_1C_0|I_{L\times L-2}]$ and so on until get $M^i=[C_{i-1}\ldots C_0|I_{L\times L-i}]$, where $1\leq I <L$.
When $C_P=C_0$ then $M^{P+1}=M$.
Now let's calculate C_i [10] s.t.
$C_i=M\times C_{i-1}, \quad i=1, 2, \ldots, r\text{-}1$ (5)
Equation (2) can be rewritten in matrix form:
$A_0\times C_i= s_i, \quad i=0,1,..,r\text{-}1$ (6)
if i=0 then $A_0\times C_0=s_0$ is the 1st equation of the LES,
if i=1 then $A_0\times C_1=s_1$ is the 2nd equation of the LES, and
if i=L-1 then $A_0\times C_{L-1}=s_{L-1}$ is the L^{th} equation of the LES.
In general:
$A_0\times\Psi=S$ (7)
Ψ represents the matrix of all C_i vectors s.t.
$\Psi = (C_0C_1\ldots C_{L-1})$ (8)
The LES can be formulated as follows:
$A = [\Psi^T|S^T]$ (9)
A represents the extended (augmented) matrix of the LES.

Example (1)

Let the SR_3 has $C_0^T=(0,1,1)$ and S=(0,0,1), by using equation (5), we get:

$$C_1=M\times C_0=\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \text{ in the same way, } C_2=\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

From equation (7) we have:

$A_0\begin{pmatrix}0 & 1 & 1\\ 1 & 1 & 1\\ 1 & 0 & 1\end{pmatrix}$ = (0,0,1), this system can be written as equations:

$a_{-2}+ a_{-3} = 0$
$a_{-1}+ a_{-2} = 0$
$a_{-1}+ a_{-2}+ a_{-3}= 1$

Then the augmented matrix A of LES after using formula (9) is:

$$A=\left[\begin{array}{ccc|c}0 & 1 & 1 & 0\\ 1 & 1 & 0 & 0\\ 1 & 1 & 1 & 1\end{array}\right] \qquad (10)$$

3.2.2 NLES for Brüer Cryptosystem

Now we apply this construction process for Brüer Cryptosystem, using equations (5) and (6). The CF of this generator is [8]:

$F(x_1, x_2, x_3) = x_1x_2 \oplus x_1x_3 \oplus x_2x_3$ (11)

for this reason $r=L_1L_2+L_1L_3+L_2L_3$.

The initial value of this cryptosystem is:

$A_0=A_{01}A_{02}+A_{01}A_{03}+A_{02}A_{03}=(d_0,d_1,\ldots,d_{r-1})$ (12)

(where + is concatenation to the vectors) s.t.

$d_0=a_{-11}a_{-12}$, $d_1=a_{-11}a_{-22},\ldots$, $d_{r-1}=a_{L_2 2}a_{L_3 3}$, or it can be taken from the following equation:

$$d_k=\begin{cases}a_{-i1}a_{-j2}, \text{when } \ k=i*L_2+j, \text{s.t. } \ i=0,\ldots,L_1-1, \ \ j=0,\ldots,L_2-1\\ a_{-i1}a_{-j3}, \text{when } \ k=i*L_3+j+L_1L_2, \text{s.t. } \ i=0,\ldots,L_1-1, \ \ j=0,\ldots,L_3-1\\ a_{-i2}a_{-j3}, \text{when } \ k=i*L_3+j+L_1L_2+L_1L_3, \text{s.t. } \ i=0,\ldots,L_2-1, \ \ j=0,\ldots,L_3-1\end{cases} \qquad (13)$$

(this arrangement of unknowns can be changed according to the researcher requirements so it is not standard).

In the same way, equation (13) can be applied on the feedback vector C_{ij}:

$C_i=C_{i1}C_{i2}+C_{i1}C_{i3}+C_{i2}C_{i3}$.

And the sequence S will be:

$S=S_1S_2+S_1S_3+S_2S_3$ s.t. $s_i=s_{i1}s_{i2} \oplus s_{i1}s_{i3} \oplus s_{i2}s_{i3}$, where s_i is the element i of S.

So the NLES which be changed to LES can be gotten by equation (12).

Figure (2) shows the sequence S which is generated from Brüer Generator [10].

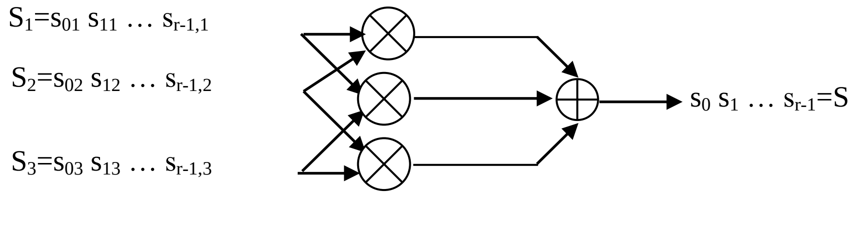
$S_1=s_{01}\ s_{11} \ldots s_{r-1,1}$
$S_2=s_{02}\ s_{12} \ldots s_{r-1,2}$
$S_3=s_{03}\ s_{13} \ldots s_{r-1,3}$
$s_0\ s_1 \ldots s_{r-1}=S$

Figure 2-The output sequence S generated from Brüer Cryptosystem

Example (2)

Let's have the following feedback vectors for 3 LFSR with length 2, 3 and 4:

$C_{01}=\begin{pmatrix}1\\1\end{pmatrix}$, $C_{02}=\begin{pmatrix}1\\0\\1\end{pmatrix}$ and $C_{03}=\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$, then r=2*3+2*4+3*4=26.

And let the required sequence is:
S=(1,0,1,1,0,1,1,1,1,1,0,1,1,0,1,0,0,1,1,0,0,1,0,1,0,1,1,0).
By using equation (5),

$C_{01}=C_{31}=C_{61}=C_{91}=C_{12,1}=C_{15,1}=C_{18,1}=C_{21,1}=C_{24,1}=\begin{pmatrix}1\\1\end{pmatrix}$,

$C_{11}=C_{41}=C_{71}=C_{10,1}=C_{13,1}=C_{16,1}=C_{19,1}=C_{22,1}=C_{25,1}=\begin{pmatrix}0\\1\end{pmatrix}$,

$C_{21}=C_{51}=C_{81}=C_{11,1}=C_{14,1}=C_{17,1}=C_{20,1}=C_{23,1}=\begin{pmatrix}1\\0\end{pmatrix}$.

$C_{02}=C_{72}=C_{14,2}=C_{21,2}=\begin{pmatrix}1\\0\\1\end{pmatrix}$, $C_{12}=C_{82}=C_{15,2}=C_{22,2}=\begin{pmatrix}1\\1\\1\end{pmatrix}$, $C_{22}=C_{92}=C_{16,2}=C_{23,2}=\begin{pmatrix}0\\1\\1\end{pmatrix}$, $C_{32}=C_{10,2}=C_{17,2}=C_{24,2}=$

$\begin{pmatrix}1\\1\\0\end{pmatrix}$,$C_{42}=C_{11,2}=C_{18,2}=C_{25,2}=\begin{pmatrix}0\\0\\1\end{pmatrix}$,$C_{52}=C_{12,2}=C_{19,2}=\begin{pmatrix}0\\1\\0\end{pmatrix}$, $C_{62}=C_{13,2}=C_{20,2}=\begin{pmatrix}1\\0\\0\end{pmatrix}$.$C_{03}=C_{15,3}=\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$

,$C_{13}=C_{16,3}=\begin{pmatrix}1\\0\\1\\1\end{pmatrix}$,$C_{23}=C_{17,3}=\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$,$C_{33}=C_{18,3}=\begin{pmatrix}0\\1\\1\\1\end{pmatrix}$, $C_{43}=C_{19,3}=\begin{pmatrix}1\\1\\1\\0\end{pmatrix}$,$C_{53}=C_{20,3}=\begin{pmatrix}0\\1\\0\\1\end{pmatrix}$, $C_{63}=C_{21,3}=\begin{pmatrix}1\\0\\1\\0\end{pmatrix}$,

$C_{73}=C_{22,3}=\begin{pmatrix}1\\1\\0\\1\end{pmatrix}$, $C_{83}=C_{23,3}=\begin{pmatrix}0\\0\\1\\1\end{pmatrix}$, $C_{93}=C_{24,3}=\begin{pmatrix}0\\1\\1\\0\end{pmatrix}$,$C_{10,3}=C_{25,3}=\begin{pmatrix}1\\1\\0\\0\end{pmatrix}$, $C_{11,3}=\begin{pmatrix}0\\0\\0\\1\end{pmatrix}$, $C_{12,3}=\begin{pmatrix}0\\0\\1\\0\end{pmatrix}$, $C_{13,3}=\begin{pmatrix}0\\1\\0\\0\end{pmatrix}$,

$C_{14,3}=\begin{pmatrix}1\\0\\0\\0\end{pmatrix}$.

by applying equation (5), C_0^T will be:
C_0^T=(1,0,1,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,0,0,0,0,1,0,0,1).
Therefore the augmented matrix will be:

$$A=\left[\begin{array}{cccccccccccccccccccccccccc|c}1&0&1&1&0&1&1&0&0&1&1&0&0&1&1&0&0&1&0&0&0&0&1&0&0&1&1\\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\ 0&0&0&0&0&1&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\end{array}\right] \qquad (14)$$

3.3 Cipher Only Attack

In this type of attack we may need some kinds of soft computing techniques. These techniques based on individuals (or population of individuals), which are considered as random solutions for the discussed problem. For cryptanalysis of stream cipher problem the individuals mean the initial state of the combined LFSR's, then generate an output key (K_i). The observed key K_i xored with available ciphertext C_i to obtain the P_i. We notice that the probability of zero's is more than 0.6, so we can use this benefit to obtain the actual key. The details of this attack mentioned in section 4.2.

4. Proposed Cryptanalysis System for Stream Cipher Cryptosystems

The proposed cryptanalysis attack is related to the amount of the available information, in another word, it's either plaintext attack or cipher only attack. Figure-3 shows the block diagram for implementing of the proposed cryptanalysis system for stream cipher cryptosystems.

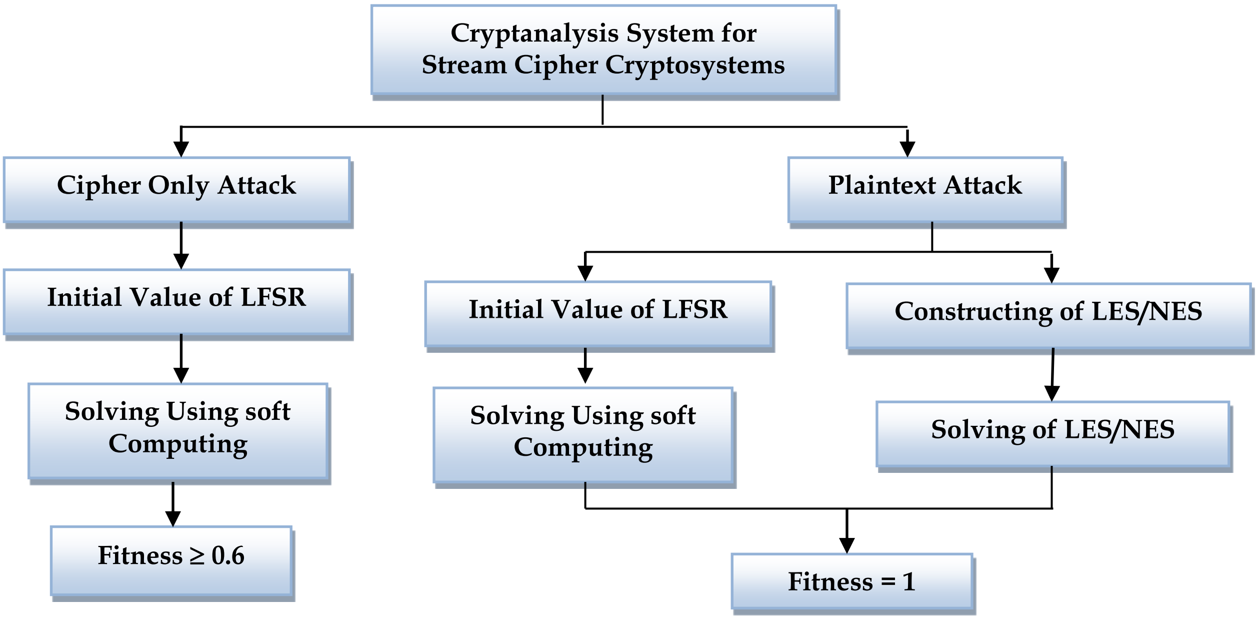

Figure 3- The block diagram of the proposed cryptanalysis system for stream cipher.

4.1 Cryptanalysis Using Plaintext Attack

For this type of attack we can solve the LES (9) by using any solver for LES, like Gauss elimination, Gauss-Jordan, etc… For example, when solving the LES (10), we obtain that A_0=(1,1,1).

While if we try to cryptanalysis a stream cipher system by solving the LES using soft computing techniques, we have to establish a fitness value by applying the following steps:

1. From population, an individual initial string of length n-bits extended to r bits m=n=L for Single LFSR and for the Brüer cryptosystem $r=L_1*L_2+L_1*L_3+L_2*L_3$, so we get the string $X=(X_1, X_2,\ldots, X_r)$ after extension.
2. The extended string bit X_j product with corresponding equation string bit Y_j, where $1\leq j\leq r$ s.t. the equation string is $Y=(Y_1,Y_2,\ldots,Y_r)$ and calculate the observed value:

$$O_i=X_1*Y_1\oplus X_2*Y_2\oplus\ldots\oplus X_r*Y_r=\sum_{j=1}^{r} X_j * Y_j \qquad \ldots \qquad (15)$$

3. Compare the observed value O_i with key value K_i which represents the known output value of the cryptosystem, by using mean absolute error (MAE) s.t.

$$\text{MAE} = \frac{1}{r}\sum_{i=1}^{r}\left|O_i - K_i\right| \qquad (16)$$

4. The Fitness value is

$$\text{Fitness} = 1\text{-MAE} = 1\text{-}\frac{1}{r}\sum_{i=1}^{r}\left|O_i - K_i\right| \qquad (17)$$

where
r : The length of the individual string or equation string.
X_j: is the initial value j in String X.
Y_j: is the equation variable j in the string Y.
O_i: is the measured or observed value i calculated from equation (15).
K_i: is the key bit (actual value) i.

When the observed value O_i matches the key bit K_i, for all $1 \leq i \leq r$, then the summation terms MAE in equation (16) evaluate to 0 so the fitness value is 1. The fitness equation is bounded below by 0 though it does not actually evaluate to 0. The fact that a fitness value of 0 is never achieved does not affect the algorithm since high fitness values are more important than low fitness values. As a result, the search process is always moving towards fitness values closer to or equal 1. The steps of the

Fitness Algorithm are shown below:

Fitness Function Algorithm

```
INPUT      : READ X vector; {Initial string with length L}
             READ Y vector; {Equation string from data base file}
             READ K vector;{Actual key=absolute value of LES}
OUTPUT   : Fitness value;
PROCESS  : FOR i = 1 : r
                $O_i = \sum_{j=1}^{r} X_j * Y_j$ ; {XOR sum, $O_i$ is observed key}
Difi = |Oi-Ki|;
           END;
                $MAE = \frac{1}{r}\sum_{i=1}^{r} Dif_i$ ;{ MAE is the Mean Absolute Error}
                Fitness = 1-MAE;
END.
```

4.2 Cryptanalysis Using Ciphertext Only Attack

In this type of attack, no more need for constructing LES or NES because the actual key is no more be available. So here we depend on soft computing only, to cryptanalysis the stream cipher cryptosystems. So we have to suggest a fitness value suitable to ciphertext only attack. The proposed fitness value exploits the plaintext coding weakness when using weak coding system, like ASCII code. We note that for English language, the redundancy of E, T, A, S, I,… are high compared with Q, K and Z. In English plaintext we notice that the probability (P_0) of 0's is more than 0.6.

Let n_1 be number of 1's for a binary sequence with length r, directly n_0=r-n_1, then:

$$P_0 = \frac{n_0}{r} = 1 - P_1 \qquad \dots \qquad (18)$$

Where P_1 is the probability of 1's.

Of course in cipher only attack all we have is the ciphertext (C_i), if we obtain any output key (K_i) for any individual when using soft computing then, by using equation (18), the fitness value is calculated as follows:

$$\text{Fitness}=1-\frac{1}{r}\sum_{i=1}^{r}(C_i \text{ xor } K_i) \qquad \dots \qquad (19)$$

When the fitness value more than 0.6, this mean we obtain the actual key, otherwise we search for better key.

5. Conclusions

This research concludes the following aspects:

1. As a logical mathematical situation, for plaintext attack using soft computing, if the proposed system gives a fitness value less than 1.0, this mean, no results obtained so we must run the system

again, since the LES/NLES must has unique solution for fixed absolute values, no another solution gives fitness equal 1.0.

2. For cipher only attack, although the fitness value satisfied, we have to demonstrate the decrypted plaintext to guarantee that we obtain the actual key.
3. The proposed cryptanalysis system can be modified to be suitable to work on other stream cipher cryptosystems.

References

1. Konheim, A. G. **1981**. *Cryptography: A Primer*. John Wiley and Sons, Inc.
2. Yan, S. Y. **2000**. *Number Theory for Computing.* Springer-Verlag Berlin Heidelberg, New York.
3. Schneier, B. **1995.** *Applied Cryptography*. John Wiley & Sons.
4. Menezes, A., Oorschot, P. and Vanstone, S. **1996**. *Handbook of Applied Cryptography*. CRC Press.
5. Whitesitt, J. E.**1995**. *Boolean Algebra and its Application.* Dover Publications, April.
6. Brüer, J. O. **1983**. *On Nonlinear Combination of Linear Shift Register Sequences.* Internal Report LITH-ISY-I-0S72.
7. Golomb, S.W.**1982**. *Shift Register Sequences*. San Francisco: Holden Day 1967, (Reprinted by Aegean Park Press in 1982).
8. Papoulis, A.**2001**. *Probability Random Variables, and Stochastic Process.* McGraw-Hill College, October.
9. Ali, F. H.**2007**. Cryptanalysis of the Stream Cipher Systems Using the Genetic Algorithm. Proceeding of the Information Technology & National Security Conference, **1**: .2129-2171.
10. Juntao, G., Xuelian, L. and Yupu, H.**2006**. Fault Attack on the Balanced Shrinking Generator. *Wuhan University Journal of Natural Science,* **11** (6): 1773-1776.