# Algebraic Attack On Stream Cipher Based On Genetic Algorithms

**Assist.Prof.Dr. Salim Ali Abbas Alageelee** *

**Doua Muhsin Abed Ali Al Furajy***

**Muntaha Abood Jassim***

*** Al-MustansiriyahUniversity / College of Education / Computer Science Department**

## Abstract

Pseudonoise sequences generated by linear feedback shift register (LFSRs) with some nonlinear combining functions have been proposed as running key generators in stream ciphers .Genetic algorithm has become a suitable searching or optimization tool for solving many complex problems comparing with the traditional search techniques. Genetic algorithm contains many manipulations to speed up and improve the genetic algorithm performance; these manipulations are selection, crossover and mutation.

This paper considers a new approach to cryptanalysis based on the algebraic attack with the application of a directed random search algorithm called a Genetic algorithm. It is shown that such an algorithm can be used to reduce the number of trials which are needed to solve any system of linear and/or nonlinear Boolean equations and determine the initial setting (basic key) of the attacked generator using known plaintext attack, since stream cipher encryption can be expressed by a linear and / or nonlinear system of Boolean equations.

Well known system are taken for the case of study:1- Pless system,2- Geffe system, 3- Bruer system, 4- J-k flip-flop, 5- OR

system, 6- Multiplying (AND) system, 7- Police systems, and 8- Multiplexing.

Key words:-

Genetic algorithms, Algebraic attack, Single point Crossover, Linear Feedback Shift Registers (LFSRs).

Shrinking Shift Register.

**الطريقة الجبرية في تحليل أنظمة التشفير الانسيابي اعتماداً على الخوارزمية الجينية**

**ا.م.د. سالم علي عباس العجيلي** *

**م.م. دعاء محسن عبد علي الفريجي***

**م. منتهى عبود جاسم***

***الجامعة المستنصرية– كلية التربية– قسم علوم الحاسبات.**

**المستخلص**

تمتلك معظم مولدات المفاتيح الشبه العشوائية المتوفرة حالياً و المستخدمة في التشفير الانسيابي هيكلاً معيناً يكون أساسه (في العادة ) مجموعة من مسجلات الإزاحة ذات دالة التغذية المرتدة الخطية (LFSRs) ، إضافة إلى دالة ربط غير خطية function nonlinear combining مدخلاتها هي المتتابعات المتولدة من مسجلات الإزاحة و مخرجاتها تكون المفاتيح الانسيابية الشبه عشوائية.

تعتبر الخوارزمية الجينية Genetic Algorithm أداة مناسبة من طرق البحث أو طرق تحقيق ألا مثلية لحلّ العديد من المشاكل المعقّدة في الوقت الحاضر مقارنة بخوارزميات البحث التقليدية. الخوارزمية الجينية تحتوي على العديد من العمليات لتحسين وتسريع أداءها، هذه العمليات هي عملية الاختيار (Selection)، عملية التزاوج (Crossover) وعملية الطفرة (Mutation).

في هذا البحث تم بناء طريقة جديدة لتحليل هذا النوع من التشفير تعتمد بشكل أساسي على الطريقة الجبرية وباستخدام خوارزمية بحث تدعى بالخوارزمية الجينية Genetic Algorithm (GA) حيث أثبتت هذه الطريقة ومن خلال التطبيق كفاءتها العالية مقارنة بالطرق التحليلية الأخرى في تحديد الحالة الابتدائية (المفتاح الأساسي) للأنظمة المراد كسرها باستخدام نص واضح لنص مشفروذلك من خلال حل اي نظام من المعادلات البوليانية الخطية او / و اللاخطية Boolean Equations والتي تمثل طريقة توليد المفاتيح لاي نظام تشفير انسيابي.

عدد من الأنظمة المعروفة في التشفير الانسيابي تم تحليل شفرها بالطريقة أعلاه. من هذه الأنظمة

1- Pless system 2- Geffe system 3- Bruer system 4- Multiplexing 5- J-K flip-flop 6- AND (multiplying) 7- OR system 8- Police system.

## 1. Genetic Algorithms Overview

Genetic algorithms are search and optimization methods based on the mechanics of artificial selection and genetic recombination operators.

In genetic algorithms; a solution of the problem is called a collection of genes; which are simply the parameters to be optimized. A genetic algorithm creates an initial population; evaluates this population according to some criteria (fitness function), and then selects the individual according to the selection schemes, and mate (recombine) to form a new population **[1]**.

The evaluate-select-recombine sequence is repeated until one or more of the following conditions are reached: proper solution is found, time limit is reached, specific number of generations is reached and individuals in population are the same or no improvement is done on the population **[2]**.

To make the genetic algorithm work well, the user must specify the number of parameters such as the population size, selection pressure, crossover rate and mutation rate **[3]**.

The proposed system is used for cryptographic and others applications typically produce a binary sequence, that may be combined into sequences or blocks of random numbers. There are two basic classes of random generators which are deterministic and nondeterministic.

## 2. Elements of Genetic Algorithms

The genetic algorithms contain many basic elements these elements which are:

### 2.1 Encoding scheme

First step needed before applying genetic algorithms is to create a coding scheme. A coding scheme is a method for expressing a solution in a string. There is no mechanical technique for creating one **[4]**.

### 2.2 Fitness Function

To solve a problem, some means or procedures must be used to discriminate good solution from bad solution. A fitness function returns a single numerical fitness value, which is proportional to the ability, of the individual represented by that chromosome and better chromosomes are assigned higher fitness function values **[5]**.

### 2.3 Selection

During this phase of genetic algorithm, individuals are selected from the population, according to their fitness values, to produce offspring, which will make up the next generation. Good individuals will probably be selected several times in a generation; poor ones may not be selected at all. The goal of any selection method is to favor the reproduction of good individuals in the population **[6]**.

### 2.3.1 Fitness Proportionate Selection with Roulette Wheel (RWS)

This method is the most common selection method in genetic algorithms, in which the number of times, an individual is expected to be reproduced is equal to its fitness divided by the sum of all fitness in the population.

Simple method of implementing fitness proportionate selection is "Roulette Wheel" which is conceptually equivalent to give each individual a slice of a circular roulette wheel equal in area to the individual's fitness.

This selection scheme has drawback that the most significant is the possibility of premature convergence (i.e. is a situation in which the fitness variance in the population become very small, hence all individuals have similar fitness values).There is another drawback this selection scheme cannot be applied if the evaluation function can return negative value **[7]**.

### 2.4 Crossover

It is a recombinant operator that takes two individuals and combines them to form two new solutions (offspring). Crossover is not necessarily applied to all pairs of individuals selected for mating. A choice is made, depending on a probability specified by the user. If crossover is not applied; the offspring are simply duplications of the parents**[8]**.

### 2.4.1 Single Point Crossover (SPC)

Two individuals are chosen, whose chromosome strings are cuts at some randomly position, which is, called crossover point. This produces two "head" segments and two "tail" segments. The tail segments are then swapped over to produce two new full-length children **[9]**.

Chromosome A 1 **0** 1 1 0 0

Chromosome B 1 **1** 0 1 0 1

Child A 1 0 0 1 0 1

Child B 1 1 1 1 0 0

### 2.5 Mutation

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible.

Mutation is an important part of the genetic search process as it helps to prevent the population from stagnating at any local optima. Mutation occurs according to user-definable probability **[10]**.

### 2.5.1 Bits Inversion Mutation (BIM)

First order mutation is changes a single bit in a chromosome, where select bits according on a probability specified by the user from the selected chromosome and inverse it **[11]**.

Chromosome A 0 1 **0** 1 **1** 0

Chromosome A' 1 1 **1** 1 **0** 0

## 3. Stream Cipher Overview

Stream ciphers are an important class of encryption algorithms that encrypts individual characters (usually binary digits) of a plaintext message one at a time, when a block ciphers tend to simultaneously encrypt groups of characters of a plaintext message using a fixed encryption transformation. Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry.

They are also more appropriate, when buffering is limited or when characters must be individually processed as they are received, stream ciphers may also be advantageous in situations where transmission errors are highly probable **[11]**, **[12]**.

There is a vast body of theoretical knowledge on stream ciphers, and various design principles for stream ciphers have been proposed. This unfortunate state of affairs can partially be explained by the fact that most stream ciphers used in practice tend to be proprietary and confidential **[13]**.

## 3.1 Stream Cipher Classification [14], [15]

The stream ciphers process plaintext in blocks as small as a single bit, and the encryption function may vary as plaintext is processed; thus stream ciphers are said to have memory. They are sometimes called state ciphers since encryption depends on not only the key and plaintext, but also on the current state. Stream ciphers can be either symmetric-key or public-key. As illustrate below.

### 3.1.1 The one-time pad stream cipher [16]

Vernam cipher is a good example on a one time pad over the binary alphabet this algorithm can defined by the equation (1)

$$C_i = M_i \oplus K_i \qquad \textbf{Equation (1)}$$

Where $M_i$ are the plaintext digits, $K_i$ are the key digits, $C_i$ are the ciphertext digits and $\oplus$ are the XOR function are used. If the key stream digits are generated independently and randomly, the Vernam cipher is called a one-time pad, and is unconditionally secure against a ciphertext only attack, see Figure (1).

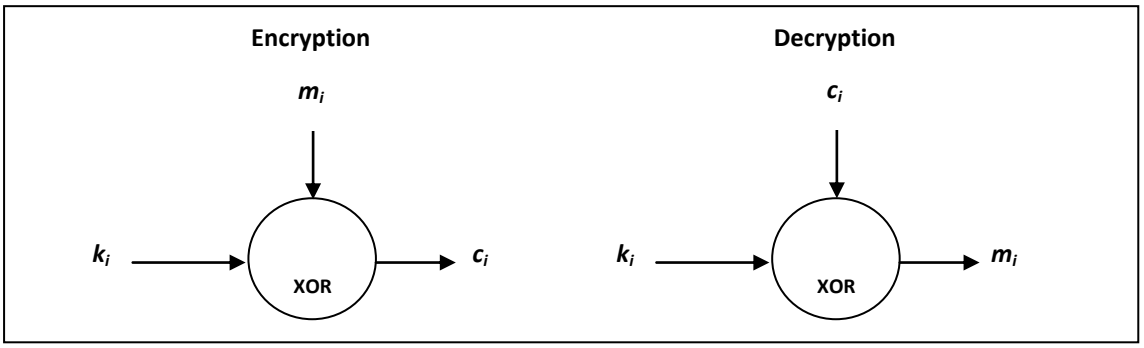



**Figure (1) General model of a one-time pad stream cipher**

Shannon proved that a necessary condition for a symmetric key

Shannon proved that a necessary condition for a symmetric key regardless of the statistical distribution of the plaintext, and is optimal in the sense that its key is the smallest possible among all symmetric-key encryption schemes having this property.

An obvious drawback of the one-time pad is that the key should be as long as the plaintext, which increases the difficulty of key distribution and key management.

### 3.1.2 Synchronous stream cipher [17]

A synchronous stream cipher is one in which the key stream is generated independently of the plaintext message and of the ciphertext. The encryption process of a synchronous stream cipher can be described by the equations (2)

$$\left.\begin{aligned} p_{i+1} &= f(p_i, k) \\ z_i &= g(p_i, k) \\ c_i &= h(z_i, m_i) \end{aligned}\right\} \qquad \textbf{Equation (2)}$$

Where $p_i$ is the initial state and may be determined from the key $k$, $f$ is the next-state function, $g$ is the function which produces the key stream $z_i$, and $h$ is the output function which combines the key stream and plaintext $m_i$ to produce ciphertext $c_i$, In a synchronous stream cipher, both the sender and receiver must be synchronized using the same key to allow for proper decryption. The ciphertext digit that is modified during transmission does not affect the decryption of other ciphertext digits, the encryption and decryption processes are depicted in Figure (2).

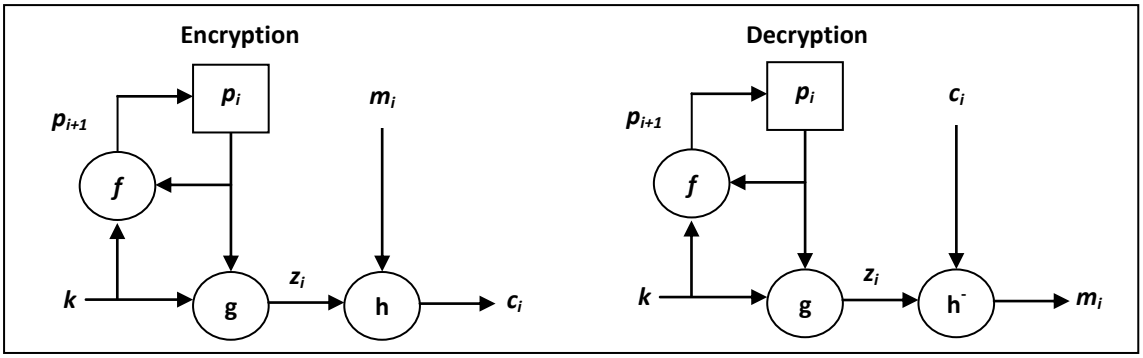



**Figure (2) General Model of a Synchronous Stream Cipher**

### 3.1.3 Asynchronous stream cipher [18]

Asynchronous stream cipher is one in which the key stream is generated as a function of the key and a fixed number of previous ciphertext digits. The encryption function of asynchronous stream cipher can be described by the equation (3)

$$\left.\begin{aligned} p_i &= (c_{i-1}, c_{i-2}, c_{i-3}, c_{i-t}) \\ z_i &= g(p_i, k) \\ c_i &= h(z_i, m_i) \end{aligned}\right\} \quad \textbf{Equation (3)}$$

Where $p_i = (c_{i-1}, c_{i-2}, c_{i-3}, c_{i-t})$ is the (non-secret) initial state, $k$ is the key, $g$ is the function which produces the key stream $z_i$ and $h$ is the output function which combines the key stream and plaintext $m_i$ to produce ciphertext $c_i$. The encryption and decryption processes are illustrated in Figure (3).

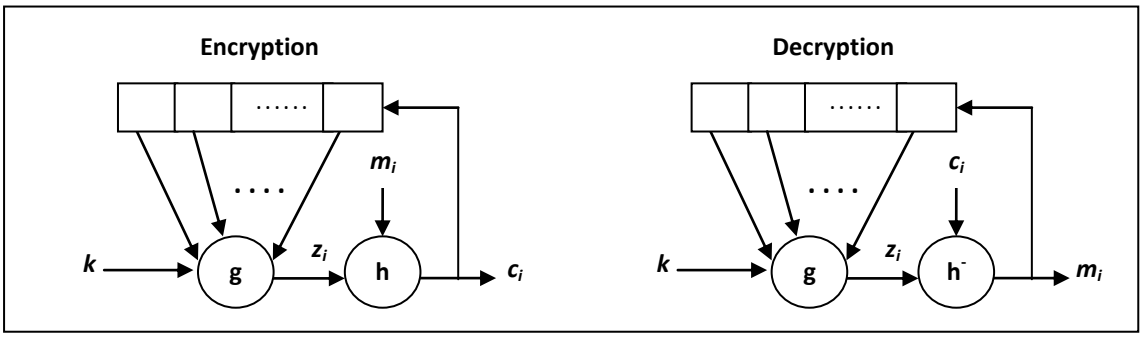



**Figure (3) General model of asynchronous stream cipher**

Asynchronous is possible if cipher text digits are deleted or inserted, because the decryption mapping depends only on a fixed number of preceding ciphertext characters. Such ciphers are capable of re-establishing proper decryption automatically after loss of synchronization.

Suppose that the state of asynchronous stream cipher depends on previous ciphertext digits. If a single ciphertext digit is modified during transmission, then decryption of up to subsequent ciphertext digits may be incorrect, after which correct decryption resumes, since each plaintext digit influences the entire following ciphertext.

## 4. Shift Register [19], [20]

The vast majority of proposed key stream generators are based in some way on the use of shift registers. There are many basic methods used shift register; these methods are linear feedback shift register, nonlinear feedback shift register, feedback with carry shift register and shrinking register.

### 4.1 Linear feedback shift register (LFSR) [21]

Linear feedback shift register are very familiar to electrical engineers and coding theorists and very suited for high speed implementations since they are easily implemented in both hardware and software, they can produce sequences of large period and good statistical properties, see Figure (4).

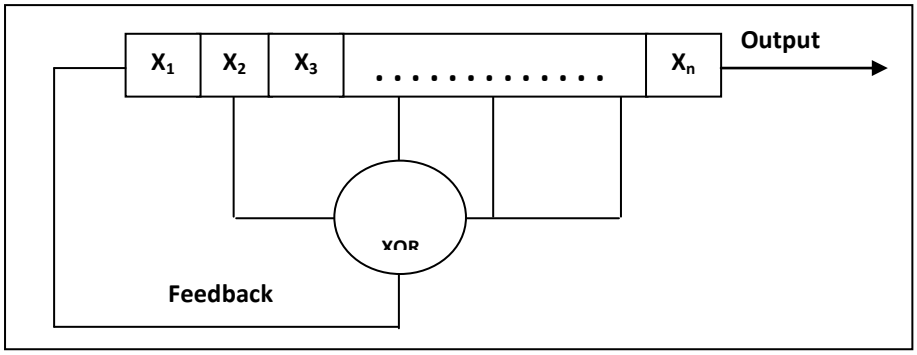



**Figure (4) Linear feedback shift register**

There are clearly has to be a drawback to such sequences that can be easily and quickly generated and seem to have good properties of random appearance, the other drawback is that they only have linear complexity since they are generated using an n-stage linear feedback shift register.

### 4.2 Nonlinear feedback shift register (NFSR) [22]

The output bit produce by a combination of Boolean function with many binary inputs, these combination should be balanced, highly nonlinear and correlation immune, see Figure (5).

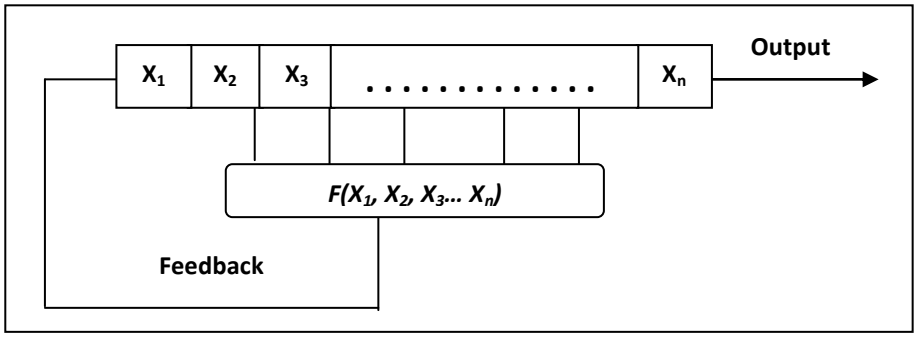



**Figure (5) Nonlinear feedback Shift register**

### 4.3 Feedback with carry shift register (FCSR) [23]

The FCSR is similar to the LFSR except that it has a small amount of auxiliary memory. The difference is that during all iterations, the memory content which is an integer is added to the sum of the selected bits and the parity of this value, see Figure (6),

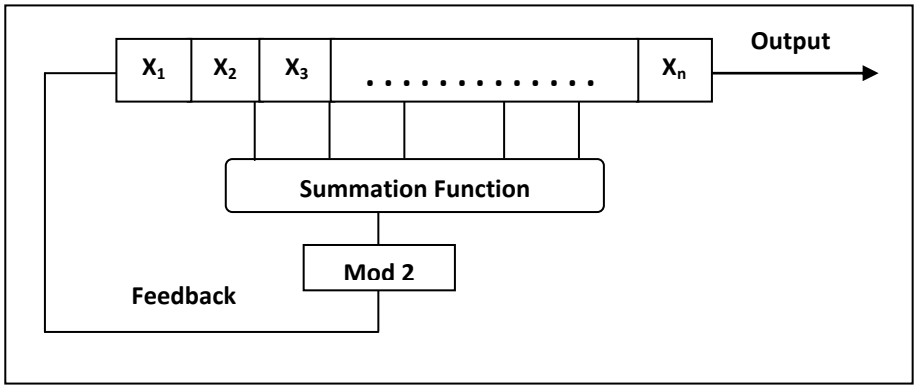



**Figure (6) Feedback with carry shift register**

### 4.4 Shrinking shift register (SSR) [24]

Shrinking shift register is a promising candidate for high speed encryption applications. The designed shift register contains a combination of (LFSR and\or NFSR and\or FCSR), then XOR function is used to produce a single bit, see Figure (7).

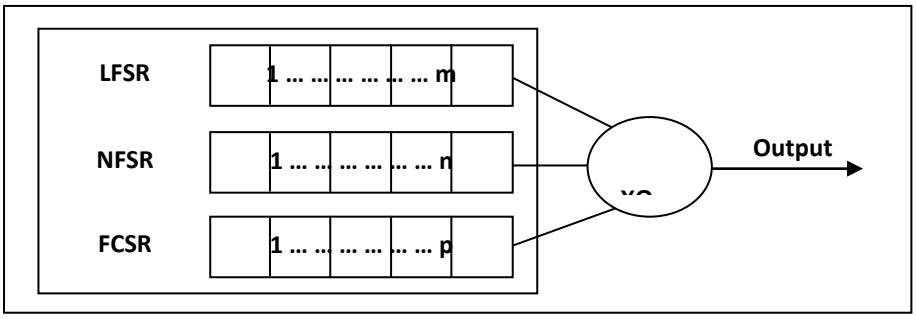



**Figure (7) Shrinking shift register**

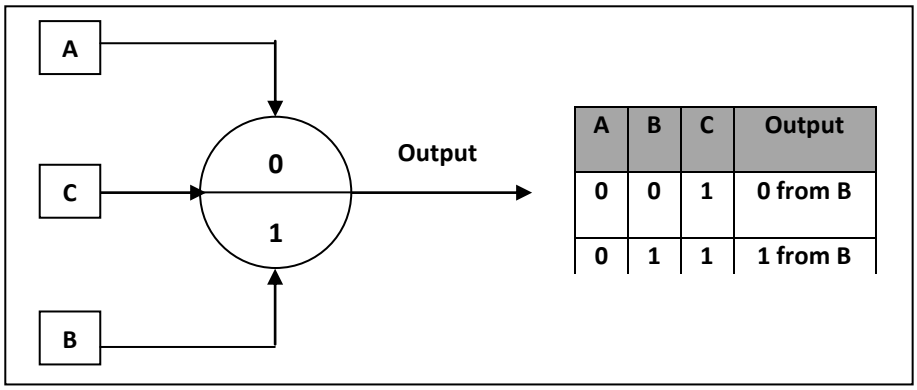
## 5. Police Circuit

A combination of logic circuits that produce a single output from three inputs. The output depending on the status of third input (C) that

determines the output. The output can be determined by the following equation (4)

$$Output = A(C+1) + B*C$$ **Equation (4)**

Below diagram explain the police circuit and a simple example on its work, see Figure (8).

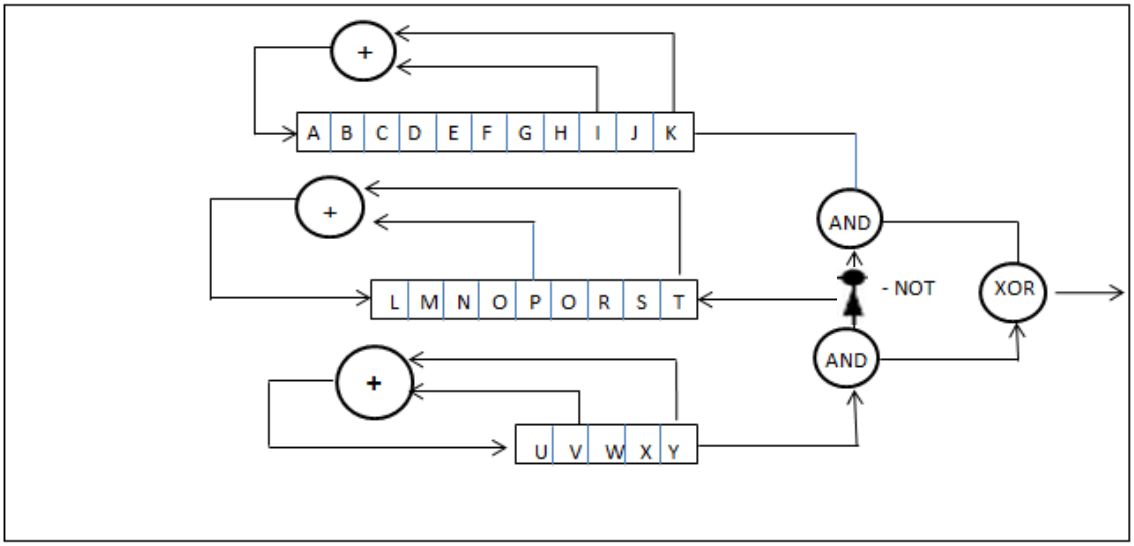



| A | B | C | Output |
|---|---|---|---|
| 0 | 0 | 1 | 0 from B |
| 0 | 1 | 1 | 1 from B |

**Figure (8) Police circuit**

## 6- The proposed system

In stream cipher encryption, the stream of binary message symbols is combined with a pseudorandom sequence by modulo 2 additions. This process can be expressed by a system of linear and / or nonlinear Boolean equations depending on the type of the combining functions which are used.




Geffe system , For instance, with shift registers lengths 11,9,5 and primitive feedback polynomials 101000000001 , 1000100001 , 100101 and initial setting symbols (A,B,C,D,E,F,G,H,I,J,K) , (L,M,N,O,P,Q,R,S,T), (U,V,W,X,Y) respectively as show in figure 9:

Figure (9)Geffe system with shift registers length 11,9,5

Can be expressed by a system of nonlinear Boolean equations such as:

K*(T+1) + T*Y=K1

K*(S+1) +S*X=K2

I*(R+1) +R*W=K3

H*(Q+1) +Q*V=K4

G*(P+1) +P*U=K5

F*(O+1) +O*(Y+V) =K6

E*(N+1) +N*(X+U) =K7

: : : : :

: : : : :

: : : : :

This system of nonlinear Boolean equations describes the way in which Geffe algorithm works.

In a known plaintext attack, the known plaintext is XORed with the corresponding cipher text to find the corresponding key stream sequence Ki (i=1, 2, 3,…) which is the output of the key generator i.e the right hand side of the Boolean equations.

One way for solving such system of Boolean equations is to try all possible solutions (key space) that's mean if the system of Boolean equations has n variables, we have to try $2^n$ -1possible solutions to find the correct variables values.

The above system of nonlinear Boolean equations of Geffe example has 25 variables, and hence, we have to try $2^{25}$ -1 = 32554431 possible solution, to determine the correct initial setting of Geffe system.

In this section we present a new approach for solving any system of Boolean equations (linear and / or nonlinear) based on the application of a directed random search algorithm, called Genetic Algorithm.

It is shown that such algorithm can be used to reduce the number of possible solutions (key space) to be searched to solve any system of Boolean equations and determine the correct initial setting (symbols values) of the attacked generator .

**6-1Application of Genetic Algorithm (GAs) to key search**.

The focus of this section is to apply a genetic algorithm to the problem of searching through the key space of stream cipher systems which are expressed by systems of linear and \ or nonlinear Boolean equations.

There are four major steps in preparing to use the conventional genetic algorithm on fixed-length character strings to solve a problem. These four major steps need to be defined in order in order to set up such an approach.

**6-1-1 Key representation (coding scheme).**

A coding scheme is a method for expressing a solution in a string.

Type of coding scheme to use depends on the problem. The coding scheme we have chosen is given by a sequence of 0's and 1's (binary string).the length of the string represents the number of distinct variables in the system of equations.

The order of 0 and 1 in the binary string represents the underlying substation . this means , the key 10110001110 indicates that the number of variables in the system of Boolean equations is 11 (eleven) and the variables (A,B,C,D,E,F,G,H,I,J,K) values are going to be substituted in the system of Boolean equations are :

A=1,B=0,C=1,D=1,E=0,F=0,G=0,H=1,I=1,J=1,K=0.

**6-1-2 Creating a Fitness Function**

In this study, our purpose is to find the (Binary) string which satisfies all the given equations. The fitness function selected for this study is based on the number of equations which are satisfied by a given binary string.

**Fitness = number of equations which are satisfied by a given binary string**.

This process is as follows:-

1- A given string is used to substitute in the system of equations.

2- The number of equations which are satisfied are counted and assigned as fitness value.

**6-1-3 Termination Conditions**.

The threshold value (T) which was chosen for this study was the number of equation which was created using the attacked generator.

T= Number of equations which are variable in the system file.

We have used thus value (T) as a termination condition to a run of our genetic system, and this was done by comparing T value with the performance of all binary strings (fitness value) of all population strings (structures) in each generation. We also have used a number of generations to be run as another termination criterion.

**6-1-4 the mating process**

The breeding process (crossover operator) itself is achieved using single point crossover method which was described above.

**6-1-5 the Mutation process**

The mutation operator is achieved using the bit inversion mutation method which was described above.

**6-1-6 The complete Algorithm**

These processes are combined to create the complete genetic algorithm.

The steps of the algorithm are:-

1- A random population of binary string is generated.

2- A fitness value for each string in the population is determined.

3- A biased random selection of parents is considered (based on fitness)

4- The crossover operation is applied.

5- The mutation process is applied to the children.

6- A fitness value for each string in the new generation is determined.

**The above algorithm can be written as**:

Input:

POPSIZ [Population size]

CHROMLEN [chromosome length]

EQFILE [equation file name]

MAXGEN [Maximum number of generation]

PRCROSS [crossover probability]

PRMUTATE [Mutation probability]

THRESHOLD (T) [Threshold value]

RANDOM SEED [Seed random number]

Process:

t ← 0 // t is the generation number //

Initialize p (t)

Evaluate strings in p (t)

While (t< MaxGen) and (fitness < Threshold) Do

t←t+1

Select p (t) from p (t-1)

Recombine strings in p (t)

Evaluate strings in p (t)

Repeat

Output:

Initial setting [initial setting of the attacked generator].

This process will stop after a fixed number of generation or a specified value (T) will be met.

**6-2Results**

The simulation of this algorithm is programmed in Pascal. It was applied to many systems of linear and nonlinear Boolean equations. The following table (1) shows some samples of those generators which were attacked using genetic method. It also shows the number of equations which are needed and the input parameters to genetic system and generation number which contains the correct setting (initial setting) of the attacked generator.

| No.Of S.R | S.R lengths | Combining Fn.Type | No.Of Eq. | Key Space | Cr. Prob. | Mu. Prob. | Pop. size | Gen. No. |
|---|---|---|---|---|---|---|---|---|
| ٢ | 3,4 | J-K F-F | ١٤ | ١٢٧ | 0.80 | 0.05 | ١٠ | ١ |
| ٢ | 5,4 | AND | ١٨ | ٥١١ | 0.80 | 0.05 | ١٠ | ١ |
| ٣ | 4,3,2 | GEFFE SYS. | ١٨ | ٥١١ | 0.80 | 0.05 | ١٠ | ٢ |
| ١ | ١١ | POLICE | ١١ | ٢٠٤٧ | 0.70 | 0.05 | ١٠ | ٤ |
| ٢ | 5,7 | OR | ١٢ | ٤٠٩٥ | 0.80 | 0.05 | ١٠ | ١٦ |
| ٢ | 7,11 | J-K F-F | ١٨ | ٢٦٢١ ٤٣ | 0.75 | 0.05 | ١٨ | ٣٣ |
| ٣ | 7,5,3 | GEFFE SYS | ١٩ | ٣٢٧٦ ٧ | 0.80 | 0.05 | ١٨ | ١٨ |

Table (1) shows sample of generators and their results using genetic method.

The minimum number of equations which are needed is equal to the number of variables (the sum of lengths of shift registers which are used) in the system of equations provided that all variables have to appear in

those equations. Variations on the crossover and mutation procedures may significantly affect the behavior of the algorithm.

## 6-3- Discussion

1- In this paper, it is argued that genetic algorithms are a valuable tool in the cryptanalysis of certain classes of cipher, and it is shown that stream ciphers can be broken using such a genetic algorithm.

2- This research has developed and tested the use of genetic algorithms in cryptanalysis of stream cipher. This was achieved the proposed system.

3-Our method have been applied to different samples of stream cipher system, and they proved highly successful in finding a key of the attacked generators. Their results are fully described table (١).

4- Optimization of control parameters for genetic algorithm in our methods has been made.

## 6-4 Conclusion

1-It is pointed out that the analysis of our method (cryptanalysis with a known-plaintext attack) is valid for all stream cipher systems.

2- Our original goal was clearly met. The genetic algorithm proved highly successful in cryptanalysis of a stream cipher systems and determining the initial setting of the attacked generators.

3- This work suggests that a new approach is needed to characterize problem that may be difficult for genetic algorithm

4- Genetic algorithm can be used as a powerful tool in generating a pseudo random sequences with good statistical properties and a high linear complexity and overcome all problems and difficulties which face designers of cipher systems.

## 7-Future works

Several areas for future work suggest themselves:

(1) Investigating the noise of selection, the noise of genetic operators, and the explicit noise or nondeterminism of the objective function and their effect on the cryptanalysis of cipher systems.

(2) Numerous modifications have been made to the conventional genetic algorithm. One of those modifications is called steady-state genetic algorithm, another simple modification which has been made to the conventional genetic algorithm is the replacement of linear crossover with a cyclic crossover scheme. Investigating the effect of those modifications on the cryptanalysis of cipher systems which were attacked using the conventional genetic algorithm.

(3) Extra, or different, fitness functions might be investigated which may give better results than those reported here.

(4) In addition, other more complicated ciphers such as block cipher may be analyzed, using genetic algorithm.

(5) Use of parallel genetic algorithm in the cryptanalysis of stream cipher system might be investigated.

## 8. References

1. Goldberg D., "*Genetic Algorithm in Search, Optimization and Machine Learning*", 1989, Addison Wesley Longman, USA.
2. Whilley L. and Vase M., "*Foundations of Genetic Algorithms*", 1995, Morgan Kaufmann Publishers, USA.
3. Sabah M., "*A Comparative Study between Traditional Genetic Algorithms and Breeder Genetic Algorithms*", 2004, M.Sc., Thesis, AL-Nahrain University.
4. Mitchell M., "*An Introduction to Genetic Algorithms*", 1996, MIT Press, England.
5. Koza J., "*Genetic Programming: On the Programming of Computers by Means of Natural Selection*", 1993, MIT Press, England.
6. Winter G., Periaux J. and Galan M., "*Genetic Algorithms in Engineering and Computer Science*", 1995, John Wiley and Sons, USA.
7. Michalewicz Z., "*Genetic Algorithms + Data Structures = Evaluation Programs*", 1996, Springer-Verlag, Germany.
8. *De Jong K. and Spears W.*, "*A Formal Analysis of the Role of Multi-point Crossover in Genetic Algorithms*", Annals of Mathematics and Artificial Intelligence Journal, 1992, Scientific Publishing Company, Switzerland.
9. Liepins G. and Vose M., "*Adaptation and genetic algorithms*", 1991, Morgan Kaufmann Publisher, USA.
10. Winter G., Periaux J. and Galan M., "*Genetic Algorithms in Engineering and Computer Science*", 1995, John Wiley and Sons, USA
11. Bartosz Z., "*One-Way Function and Stream Cipher*", First Edition, P/19, IACR Press, India, 2004.
12. Avinash K., "*Block and Stream Ciphers in Real-World Systems for Secure Communications*", Technical Report, P/26, Computer and Network Security School, University of Purdue, USA, 2010.
13. Robshaw M., "*Stream Ciphers and RSA*" Technical Report, P/71, RSA Laboratories, University of California, USA, 1995.
14. Menezes A., Oorschot P., and Vanstone S., "*Handbook of Applied Cryptography*", First Edition, P/191-212, CRC Press, USA, 1997.
15. Schneier B., "*Applied Cryptography*", Second Edition, P/390-398, John Wiley and Sons, USA, 1996.

16. Hoon-Jae L., "*Introduction to Stream Cipher (Past, Present and Future)*", Technical Report, P/39, Cryptography and Network Security Laboratories, University of Dongseo, 2010.
17. Michalis G., Paris K., Kostopoulos G., Nicolas S. and Goutis C., "*Comparison of the Hardware Implementation of Stream Ciphers*", Second Edition, P/268, Hellenic Press, Greece, 2005.
18. Joan D. and Paris K., "*Self-Synchronizing Stream Cipher*", Second Edition, P/28, Hellenic Press, Greece, 2005.
19. Anashin V., Andrey B. and Kizhvatov I., " *New Fast Flexible Stream Cipher*", First Edition, P/29, Springer-Verlag, Russian, 2006.
20. Wagstaff S., "*Prime Numbers with a Fixed Number of One Bits or Zero Bits in Their Binary Representation*", Lecturer Notes, P/267-273, Institute for Experimental Mathematics, Germany, 2001.
21. Akio T., Sho M. and Takahiro I., "*A Study on Generation of Random Bit Sequences with Post-Processing by Linear Feedback Shift Registers*", Proceeding In International Journal of Innovative Computing, P/2631-2638, China, 2008.
22. Lan L., QiongHai D., ZhiGuang Q. and ChunXiang X., "*Intelligent Stream Cipher Fuse Memory Modules*", First Edition, P/20, China, 2010.
23. Arnault F., Berger T. and Necer A., "*Feedback with Carry Shift Register with the Euclidean Algorithm*", P/910-917, IEEE Transactions on Information Theory, USA, 2004.
24. Mat S. and Ahmad Z., "*Comparison Analysis of Stream Cipher Algorithms*", First Edition, P/7, Journal Technology Press, Malaysia, 2007.