Journal of Basic and Applied Engineering Research
Print ISSN: 2350-0077; Online ISSN: 2350-0255; Volume 1, Number 13; October-December 2014 pp. 43-46

http://www.krishisanskriti.org/jbaer.html

Analysis of LFSR based Cryptosystem using Genetic Algorithm

Smriti Dwivedi[1]and Maiya Din[2]

[1]Banasthali Vidyapith Jaipur, India
[2]Defence Research & Development Organization Delhi, India
E-mail: [1]dwivedismriti2@gmail.com, [2]anuragimd@gmail.com

Abstract–*In today's world, with increasing usage of computer networks and internet, information security is very important. One of the widely used approaches for information security is Cryptography. Cryptography is the branch of cryptology used for secure communication using Cryptosystems. Cryptosystems can be broadly divided into two categories stream cipher and block cipher cryptosystems. The stream cipher encrypts the data stream bit by bit. The most widely used stream ciphers are based on Linear Feedback Shift Registers (LFSRs). To analyze n-bit LFSR, the computational complexity is 2n-1. In this paper, we are analyzing LFSR based cryptosystem using Genetic Algorithm. The polynomials of degree 11, 13, 17, 19 and 23 are considered. In this approach, computational time has been reduced compare to brute force approach for finding correct initial bits of used LFSRs.*

1. INTRODUCTION

In today's world, the data or information being shared is of high value. To preserve its value the confidentiality, integrity and availability of the information must be maintained [1]. Cryptography provides security to the information, where instead of the actual message its crypt is sent. There are various cryptosystems available in the literature like stream cipher and block cipher cryptosystems [2]. A stream cipher based on shift registers transforms a bit or a byte of the plaintext at a time whereas, a block cipher encrypts a block of fixed length. Key is a most valuable asset in cryptography. Cryptanalyst requires key to break cipher text. Here we use LFSR (Linear Feedback Shift Register) to encrypt the plain text and genetic algorithm is used to find the key for deciphering the cipher text.

In 2003, Albassal et al. [3] proposed a method for finding the key of a substitution permutation network (SPN) using genetic algorithms. The fitness measure proposed used the differential characteristics of the SPN. In 2006, N. Nalini [4] analyzed attack on SDES using Optimization Heuristics technique and GA technique. It resulted in favor of GA technique because it reduced the time complexity. In 2007, Alecu, analyze the design of a genetic algorithm to k-error linear complexity of a sequence [5]. In 2007, Hasan Mohammed Hasan Husein et al. [6] adopted genetic algorithm for the improvement in cryptanalysis of DES. In 2009, Faheem Masoodi et al. [7] described the properties of LFSR like feedback period, time complexity and statistical behavior. They were analyzing the security paradigms of LFSR and different techniques for security. In this paper, we are using LFSR for encrypting the plaintext and GA for finding the key for decrypting the cipher text. We are using LFSR with primitive polynomials for encrypting the plain text. Primitive polynomial approach of LFSR makes the key more secure and difficult for the intruder. Genetic algorithm (GA) developed by John Holland [9] is a heuristic search algorithm based on natural genetics. GA uses a set of chromosomes, fitness value of chromosomes, crossover and mutation to form new off springs. To form a new generation, the fittest chromosome copied along with some new off springs. The paper is divided into six sections. Section 2 describes Linear Feedback Shift Register (LFSR) and section 3 contains a description of Genetic Algorithm. Section 4 describes the proposed approach based on GA. Section 5 summarizes and analyzes experimental results and concluding remarks are given in section 6.

2. LINEAR FEEDBACK SHIFT REGISTER

Linear Feedback Shift Register (LFSR) is used in a stream cipher to generate linear sequence of pseudo random numbers. LFSRs are also used as counters and sometimes used for error checking and correction. They require very less hardware and have high speed of operations and thus are more advantageous. An n stage LFSR is maximum length if some initial states will results a sequence that repeats every 2n - 1 bit. The distribution of patterns having fixed length is almost uniform in generated sequence. The contents of the registers are shifted by one position at each clock. The left most bit fed to the register is the result of XORing first bit and the last bit of the register. The right most bit or the last bit is used to form the pseudo random number (PN) sequence which encrypts the data. A count of all "0"s is not used as initial states because the counter would remain locked-up in these states. Generation of key stream is shown in fig 1

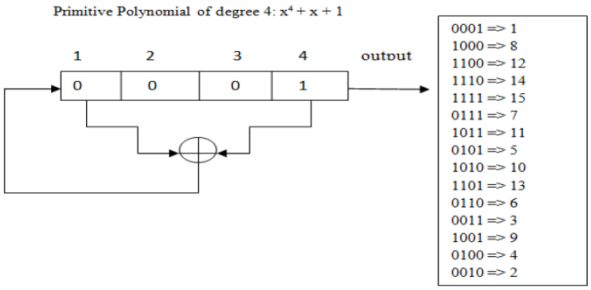

Fig. 1: Generation of key stream in LFSR.

In the given example, bits are inserted from the left end and are shifted one space towards right. The bit which is inserted on left end comes after XORing the first bit and last bit of the register. The rightmost bit becomes output bit known as key stream (PN sequence) bit. Primitive polynomials are used to generate key stream because these have a large PN period. The generated PN sequence is XORed with binary plain message to get encrypted message.

3. GENETIC ALGORITHM

Genetic Algorithm developed by John Holland [9] in 1970s is based on Darwin's theory of evolution. GA is also known as Simple Genetic Algorithm (SGA). General idea about simple genetic algorithm is that fittest chromosomes should be survived in next generation and weakest chromosomes should remove from the genetic structure. On that concept Holland derived a concept of survival for fittest solution and rare selection of weakest solution. Simple genetic algorithm has following steps in it: number of chromosomes (population or solutions), fitness function, selection method, crossover and mutation. First, number of chromosomes or solutions for a particular problem are considered and a fitness function is defined depending upon the problem. In fitness function, we have to decide whether we have to maximize or minimize the value of fittest solution. Second, choose a selection method for selecting parent chromosomes according to the fitness function (if fitness is better then, probability of selection is higher of that chromosome than other chromosomes). Different selection methods are roulette wheel selection, Boltzmann selection, tournament selection, rank selection, steady state selection etc. Third, crossover operation is applied on the parent chromosomes i.e., swapping or interchanging genes of two chromosomes. Crossover can be performed by selecting any (random) gene or bit along the length of the chromosome or polynomial and swaps the bits after crossover point. There are many kinds of crossover as: one point crossover, two point crossover, uniform crossover, arithmetic crossover etc. Fourth, mutation of the chromosomes formed by the crossover operation is done. Mutation is used to prevent the local optimum problem. The mutation probability should be less because higher probability may convert the search to random search. After mutation, if the generated off springs are more fit than the parent chromosomes, then they will be considered for the initial population for the next generation.

4. PROPOSED APPROACH

An analysis of LFSR based cryptosystem using Genetic Algorithm is proposed in this paper. LFSR is used to generate key stream bits by primitive polynomials. Generated PN sequence is used for encrypting plain text messages. The plain message is converted into binary message using Murray code. The binary plain message is XORed with the key stream bits to get encrypted message. To get plain message, we generated key stream bits and XOR them with encrypted message. To find the key for deciphering encrypted message genetic algorithm based approach is used. In this approach, initially we are considering initial bits as chromosomes to form population. Now fitness value is calculated by fitness function for each chromosome. Fitness value depends on frequencies of most frequent monograms bigrams & trigrams in English text message. The formula used for finding fitness function is:

$$Fitness = \left(\left(\sum_{i=1}^{15} F_i \times W_i \right) \times 100 \Big/ L \right)$$

Where, Fi : frequency of features as monograms, bigrams, and trigrams. Wi : Weights of first 15 monograms, bigrams, and trigrams. L : Message Length After this chromosomes are selected for mating using Roulette Wheel selection method. In Roulette wheel selection method, chromosomes are selected according to their fitness values. Better chromosomes have more chance to be selected in next population. This is done by following steps:

- Calculate the sum (s) of all chromosome fitness's.
- Generate a random number (r) between 0-s.
- When sum (s) is greater than random number then returns that chromosome.

We are using crossover on chromosomes selected for mating pool. The crossover used in it is one point crossover. In one point crossover, two polynomials are select for crossover. One bit is selected randomly and work as crossover point. Before that bit the chromosome (polynomial) remains same. After that bit the chromosome, swap its bits with the other chromosome. To get better chromosomes for next generation we apply crossover on polynomials of mating pool. Mutation is used to prevent all the solutions falling into local minima. Mutation changes or flips the bit randomly. In roulette wheel selection method, we are able to find correct initial bits up to 13 degree chromosomes. For higher degree chromosomes as 17, 19 degree chromosomes we are not able to find the correct initial bits. So, we switch method named as SGA selection method. In this method, we calculate expected count that is the fitness value of chromosome over average fitness value of the chromosomes.

Now we are able to find initial bit for 13, 17, 19 degree chromosomes. But for above degree of chromosomes as 23 degree still it is not able to find the initial bits. So, to find initial bits for higher degree, we consider fittest solutions of mutated population as well as initial population. It results initial bits for 23 degree chromosomes in minimum time complexity.

5. RESULTS

The entire scheme of analyzing LFSR based cryptosystem was implemented in MATLAB®. English text of size 400 characters was taken as the plain text. The polynomials used for encrypting the text are of degree (chromosome length) 11, 13, 17, 19 & 23. For each degree, one cipher text of size 400 characters was formed (thus total five cipher texts). Number of polynomials used in it are 30 to 60. Murray code was used for changing English text into Binary form. Probability of mutation was considered between 0.003 and 0.01. One point crossover was used to form the off springs. To reduce the time complexity, we used SGA Selection method and the population used in next generation is fittest polynomials (chromosomes) of initial population and mutated population.

5.1 Technique 1

We are using Roulette Wheel method for selection in Genetic Algorithm to find correct initial bits for 13 degree polynomial and execute it for 5000 generations and able to find correct initial bits as shown in the table:

Table 1: Solutions generated for 13 degree Generating Polynomials when Correct Solution is not Included

P_m (probability of mutation)	Chromosome length	Correct Soln. obtained in Generation	CPU Time (sec)
0.02	13	1612	153.177
0.01	13	1518	145.159
0.01	13	2835	273.361

5.2 Technique 2

We are using SGA method for selection in Genetic Algorithm to find correct initial bits for higher degree polynomials. In case of 13 degree polynomials results are:

Table 2: Correct solutions generated for 13 degree Generating Polynomials when Selection method is SGA.

P_m (probability of mutation)	Correct Soln. obtained in Generation	CPU Time (sec)	Generations
0.01	166	15.975	500
0.02	291	27.893	500
0.009	1264	119.762	2000
0.008	258	24.601	2500
0.007	888	89.139	3000
0.006	3389	320.317	4000
0.005	2359	224.813	4000
0.004	1001	95.597	4000
0.003	1228	115.668	4000

In case of 17 degree polynomials results are:

Table 3: Correct solutions generated for 17 degree Generating Polynomials.

P_m (probability of mutation)	Correct Soln. obtained in Generation	CPU Time (sec)	Generations
0.02	12477	1196.247	20000
0.01	2635	255.28	25000

We are using SGA to find initial bits of 19 degree polynomials executed for 50000 generations. Crossover point considered as 2. Therefore, results are:

Table 4: Correct solutions generated for 19 degree Generating Polynomials

P_m (probability of mutation)	Correct Soln. obtained in Generation	CPU Time (sec)
0.02	3446	1112.724
0.01	22682	7311.221
0.03	2581	878.395
0.04	2333	733.704
0.009	19651	3746.005
0.008	21857	4161.217

5.3 Technique 3

We are using SGA method for selection in Genetic Algorithm and considering best chromosomes of new population (mutated) as well as best chromosomes (having maximum fitness value) of old population (initial population)for reducing time complexity.

For 17 degree polynomials, Crossover point is considered as 2. Therefore the results for 17 degree polynomial are:

Table 5: Correct solutions generated for 17 degree Generating Polynomials when old population also included.

P_m (probability of mutation)	Chromosome length	Correct Soln. obtained in Generation	CPU Time (sec)
0.01	17	6825	688.230
0.02	17	8568	864.729

For 19 degree polynomials, Crossover point is considered as 2. Results are:

Table 6: Correct solutions generated for 19 degree Generating Polynomials when old population also included.

P_m (probability of mutation)	Correct Soln. obtained in Generation	CPU Time(sec)
0.01	29156	4999.832
0.02	5087	894.479

For 23 degree polynomial, Crossover point is considered as 4. Results are:

Table 7: Correct solutions generated for 23 degree Generating Polynomials when old population also included

P_m (probability of mutation)	Correct Soln. obtained in Generation	CPU Time (sec)
0.025	63668	13468.487
0.03	54069	11352.520
0.04	10584	2296.538
0.042	99201	21627.838

6. CONCLUSION

Cryptography is an important area for securing the data. Cryptanalysis is the way to break cipher message without having encryption key (secure key). Cryptanalysis of LFSR using GA approach with SGA as a selection method reduced the computational complexity. We used 11, 13, 17, 19 and 23 degree polynomials and analyze the LFFSR based cryptosystem using Genetic algorithm. To make it (cryptanalysis) faster we used fittest polynomials of initial population and mutated population. Thus, we conclude that SGA selection method is better than roulette wheel selection method.

REFERENCES

[1] A. Agarwal, "Secret Key Encryption Algorithm Using Genetic Algorithm," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2(4), pp. 216-218, 2012.

[2] W. Stallings, Cryptography and network Security, fourth ed.

[3] E. M. B. Albassal, "Genetic Algorithm Cryptanalysis of the basic Substitution Permutation Network," presented at the Circuits and Systems IEEE 46th Midwest Symposium, 2003.

[4] N. Nalini, "Cryptanalysis of simplified Data Encryption Standard via Optimization Heuristics," International Journal of Computer Sciences and Network Security, vol. 6(1B), 2006.

[5] Alecu, Alexandra, and A. M. Salagean, "A genetic algorithm for computing the k-error linear complexity of cryptographic sequences," presented at the Evolutionary Computation, 2007. CEC 2007.IEEE Congress on. IEEE, 2007.

[6] H. M. H. Husein, B. I. Bayoumi, F. S. Holail, B. E. M. Hasan, and M. Z. A. El-Mageed, "A Genetic Algorithm for Cryptanalysis of DES-8," International Journal of Network Security, vol. 5(2), pp. 213-219, 2007.

[7] F. Masoodi, S. Alam, and M. U. Bokhari, "An Analysis of Linear Feedback Shift Registers in Stream Ciphers," International Journal of Computer Applications, vol. 46(17), pp. 46-49, 2012

[8] I. Polak and M. Boryczka, "Breaking LFSR Using Genetic Algorithm," in Computational Collective Intelligence. Technologies and Applications, Berlin Heidelberg, 2013, pp. 731-738.

[9] J. Holland, "Genetic algorithms," Scientific american, vol. 267(1), pp. 66-72, 1992.