Cryptanalysis of Stream Cipher Cryptosystem Based on Soft Computing Techniques

Prof. Dr. Salim Ali Abbas Al-Ageel 
Riyam Noori Jwada

1 (Department of Computer Science, College of Education, Al-Mustansiriya University, Iraq)
2 (Department of Computer Science, College of Education, Al-Mustansiriya University, Iraq)

Abstract: This paper presents a new investigation for cryptanalysis stream cipher based on Genetic Algorithm (GAs), Particle Swarm Optimization (PSO), GA and PSO utilized for the automatic recovery of the key, and hence the plaintext. Based on a mathematical model, it is shown that such algorithms can be used to reduce the number of trials which are needed to determine the initial state of the attacked generator using ciphertext only attack. These algorithms have been shown to be effective at finding optimal solutions. Experimental results show the ability of GA and PSO in finding the correct secret key which is used to recover the plaintext.

Keywords: Cryptanalysis, Particle Swarm Optimization, Genetic Algorithms.

I. Introduction

Cryptanalysis is the science of recovering the plaintext of a message without access to the key. It is a method of transforming cipher text into a plaintext without knowing the key or algorithm [1][2]. However, the cryptanalysis of stream ciphers through soft computing techniques as Particle Swarm Optimization (PSOs), Genetic Algorithms (GAs) is still an emerging issue. GA is based on the evolutionary ideas of Natural selection and genetics [3]. GAs is a good candidate for the optimal solutions to optimization and search problems. The algorithm has been successfully applied to Vertex-Cover problem [4][5], Maximum-Clique problem [6][7], Regression testing [8], N-puzzle problem [9], Traveling Salesman Problem [10]. PSO was originally developed by a social-psychologist J. Kennedy and an electrical engineer R. Eberhart in 1995 and emerged from earlier experiments with algorithms that modeled the “flocking behavior” seen in many species of birds. Where birds are attracted to a roosting area in simulations they would begin by flying around with no particular destination and in spontaneously formed flocks until one of the birds flew over the roosting area [11]. PSO has been an increasingly hot topic in the area of computational intelligence. It is yet another optimization algorithm that falls under the soft computing umbrella that covers genetic and evolutionary computing algorithms as well [12].

There are many researches that have been written on using soft computing techniques to cryptanalysis different types of encryption systems some of these: A.J. Clark, in his Thesis uses various optimization heuristics in the fields of automated cryptanalysis and automated cryptographic function generation [13], M.F. Uddin in his paper focused on using of PSO in cryptanalysis of simple substitution ciphers using ciphertext only attack [14], R.R. Yako. In her research, an optimization approach such as GAs is considered to improve the cryptanalysis problem [15], S. M. Hameedin her work used PSO to cryptanalysis transposition cipher. PSO used ciphertext only attack to recover the secret [16], H.A. M. Al Sharifi, in his research focused on using of PSO algorithm to cryptanalysis stream cipher using plaintext attack choosing one Linear Feedback Shift Register (LFSR) [17], B.N. Ferriman, in his Thesis focused on the RC4 algorithm and present a new approach for cryptanalysis of the cipher by attacking RC4s state register [18], Ali A. Abd. in his research is considered a new approach to cryptanalysis stream cipher systems based on genetic algorithm (GA) [19]. The present work explores the related work done and applicability of GAs and PSO in a field of cryptanalysis. The rest of this paper has been organized as follows: Section 2 presents a brief overview of GAs. Section 3 presents a brief overview of PSOs. Section 4 presents designing cryptanalysis systems for stream cipher. Section 5 presents a comparison of results of cryptanalysis system between GA and PSO. The last section explains the conclusion.

II. Genetic Algorithm

GAs is the search heuristic that mimics the process of natural evolution [3]. It is based on the Darwin’s principle of Natural selection. According to this theory the chromosomes with the best fitness function should survive and create new offspring (survival of the fittest). GAs gives useful solution to optimization and search problem. It is a rapidly growing area of Artificial Intelligence. The GAs starts with the population which is nothing but chromosomes which can be decimal or binary or even hexadecimal. The GAs operator is applied to population in order to optimize the results [20]. The new population is formed from the old population with better fitness value. The population can be crafted using following operators:

DOI: 10.9790/0661-1901047884 www.iosrjournals.org 78 | Page
Cryptanalysis of Stream Cipher Cryptosystem Based on Soft Computing Techniques

Population size: The population size is generally taken 5 to 100 [21]. But in this paper we took it about 20,100 and 200. Numerical experiments show that too large and too small number of chromosomes in the population can lead to poor solutions [22].

Fitness Function: The fitness function decides whether the given solution is achieving the aims [23]. The proper crafting of the fitness function is a crux of the solutions. In this paper we will use the T.Siegenthaler Method as a Fitness Function[24].

Selection: The chromosomes are selected from the population for reproduction. The chromosomes with higher fitness value are more likely to be selected. [23]. There are many ways to do selection process some of them are Tournament selection, Roulette wheel selection, Stochastic universal selection, Truncation selection etcetera.

Crossover: The operator created a new offspring from the population by exchanging subsequences of two chromosomes to create two offspring. There are many types of crossovers for the binary chromosomes. Some of them are Single-point crossover, Two-point crossover, Multi-point crossover, Uniform crossover. In this work we will use the Single-point crossover which is One crossover point is selected, binary string from beginning of chromosome to the crossover point is copied from one parent, the rest is copied from the second parent [25].

Mutation: Mutation operator flips the bit in chromosomes. The purpose of mutation is to maintain the diversity within the population.

III. Particle Swarm Optimization
Swarm Intelligent is a kind of Artificial Intelligence based on the behavior of animals living in groups and having some ability to interact with another and with the environment in which they are inserted. Every particle in the swarm acts in a distributed manner using the intelligence of its own and the group intelligence. Every particle has two features: a position and a velocity. The particles exchange the information to correct their positions and velocities by using the received information [26]. In this work, GA and PSO is used as soft computing algorithm employed in cryptanalysis system to find initial values of the key stream and hence, to obtains plaintext.

3.1 Basic Elements of the PSO Technique [27][28]
The basic elements of PSO technique are briefly stated and defined as follows:
1. Particle, X: It is a candidate solution represented by an m-dimensional vector, where m is the number of optimized parameters.
2. Population, pop(t): It is a set of n particles at time i, i.e., pop(i) = [X1,..., Xn]T. The number of particles in population would be between 20to 30.
3. Swarm: It is an unsystematic moving particles population, which Band together and at the same time every particle moves in a Unorganized direction.
4. Particle velocity, V: It is the speed of the moving particles which can be characterized by k-dimensional vector.
5. Inertia weight, w: It is a control factor used to control the effect of the preceding velocities on the present velocity.
6. Individual best, p: it is the composition of the particle fitness value at the present position to the best fitness value it has ever reached.
7. Global best, p: It is the best location obtained in all individual locations.
8. Stopping criteria: it is the terms which finish search process.

3.2 PSO Methods
There are several methods of PSO depending on the shape of Updated velocity equation of the particle those are:
- Original PSO: Basic algorithm introduced to calculate the velocity and position of each particle and it is used to find the optimal solution.

velocity of individual particles updated as follows:

\[ V_{P} = V_{P} + \alpha \cdot \phi_{1} (P_{G} - X_{P}) + \alpha \cdot \phi_{2} (P_{G} - X_{P}) \]

with the position calculated as follows:

\[ X = X + V_{P} \]

Where
- \( X_{P} \): Particle position
- \( V_{P} \): Particle velocity
Cryptanalysis of Stream Cipher Cryptosystem Based on Soft Computing Techniques

P_i^1: Best position found by ith particle (personal best)
P_g^j: Best position found by swarm (global best, best of personal bests)
c_1 and c_2 : are the cognitive (individual) and social, (group) learning rates, respectively. The values of c_1 and c_2 are usually assumed to be 2. r_1 and r_2 : are uniformly distributed random numbers in the range 0 and 1.

- Inertia Weighted PSO: The inertia weighted PSO is added to decrease the velocity. Its value varies from 0.9 to 0.4. The value of the jth particle velocity can be formulated:

\[ V_{ij}^{t+1} = \omega V_{ij}^t + c_1 r_1 (p_{ij}^t - x_{ij}^t) + c_2 r_2 (p_{gj}^t - x_{ij}^t) \quad j=1,\ldots,n \quad (3) \]

Then the value of the inertia weight can be calculated:

\[ w_i = W_{max} - (W_{max} - W_{min}) / \text{imax} * i \quad (4) \]

where \( W_{max} \) is the initial value of the weight, \( W_{min} \) is the final value of the inertia weight, \( i_{max} \) is the maximum number of iterations.

In this work, Inertia Weighted PSO type is used.

IV. Designing Cryptanalysis Systems for Stream Cipher

In this work we used soft computing techniques as GA and PSO which could be implemented and applied easily to solve various optimization problems. These techniques employed for the purpose of Cryptanalysis. We suggest main steps to designing cryptanalysis systems for stream cipher, we select the Geffe generator to be attacked. These steps of the analysis and procedure can be summarized as follows:

**Step 1:** select plaintext.
**Step 2:** generate the key stream from Geffe generator system.
**Step 3:** generate the ciphertext calculated as follows:
P_i \text{ XOR } K_i = C_i when P_i = plaintext, K_i = keystream, C_i = ciphertext.
**Step 4:** design fitness function.
**Step 5:** apply soft computing techniques.
**Step 6:** select the optimal solution.

4.1 Fitness Function Calculation:

the main goal of cryptanalysis is to get the key in order to obtain the plaintext. Cryptanalysis stream cipher should get the correct key, to decrypt the ciphertext. Using soft computing techniques to cryptanalysis stream cipher needs fitness to determine the best new generation. In this work, fitness function based on correlation between \( C_n \) and \( X_n \) of the function be generated by independent and identically distributed (i.i.d) random variables (r.v.) \( \text{Xn} \) with probability distribution \( P_i \) such that \( P(X_i = 0) = P(X_i = 1) \) for all \( i \) and \( n \). The function generates i.i.d. r.v. \( Z_n = f(X_1^n, X_2^n, X_3^n, \ldots, X_s^n) \) with probability distribution \( P_i \) where \( P(Z_n = 0) = P(Z_n = 1) = 0.5 \). Further, the number of tests to find the LFSR-i-part of the key is determined as a function of the number of ciphertext digits used in the correlation attack. Let the inputs

\[ X_1^n, X_2^n, X_3^n \text{ of the function be generated by independent and identically distributed (i.i.d) random variables (r.v.) \text{Xn with probability distribution } P_i \text{ such that } P(X_i = 0) = P(X_i = 1) \text{ for all } i \text{ and } n. \]

The function generates i.i.d. r.v. \( Z_n = f(X_1^n, X_2^n, X_3^n, \ldots, X_s^n) \) with probability distribution \( P_i \) where \( P(Z_n = 0) = P(Z_n = 1) = 0.5 \). Further, the number of tests to find the LFSR-i-part of the key is determined as a function of the number of ciphertext digits used in the correlation attack.

The plaintext is assumed to be the output of a binary memoryless source (BMS) with \( P(Y_n = 0) = \text{P0} \).

The r. v. a as a measure for the correlation between \( C_n \) and \( X_n \) is where:

\[ a = \sum_{i=1}^{s} (1 - 2(C_i \text{ XOR } X_i)) = N - 2 \sum_{i=1}^{s} (C_i \text{ XOR } X_i) \quad (7) \]

Where \( Z_n \) is the output key of random generator number \( n \).

The plaintext is assumed to be the output of a binary memoryless source (BMS) with

\[ P(Y_n = 0) = \text{P0} \quad (6) \]

The r. v. a as a measure for the correlation between \( C_n \) and \( X_n \) is where:

\[ a = \sum_{i=1}^{s} (1 - 2(C_i \text{ XOR } X_i)) = N - 2 \sum_{i=1}^{s} (C_i \text{ XOR } X_i) \quad (7) \]

Where \( Z_n \) is the output key of random generator number \( n \).

DOI: 10.9790/0661-1901047884 www.iosrjournals.org 80 | Page
Here, the best fitness?= 0.60, this rate considered the threshold in our work, it change according to different plaintext size.

We will attack the Geffe generator, nonlinear combining function which it consist of 3 LFSR in different Length:7,9,11.

The algebraic normal form is:

\[ f(x_1, x_2, x_3) = x_1 x_2 \text{ XOR } x_2 x_3 \text{ XOR } x_3. \]

Here, the number of ciphertext symbols is determined to perform a ciphertext-only attack on the Geffe Cipher using the correlation attack. Our conclusion from the analysis is that the pseudonoise generator's output sequence and the sequences generated by the linear feedback shift registers should be uncorrelated. This leads to constraints for the nonlinear combining function to be used.

4.2 Using Genetic Algorithm (GA) to cryptanalysis stream cipher systems

GA has been successfully applied to numerous applications in the field of search and optimization. It is recursive procedure that consists of a fixed population size of chromosomes. These chromosomes are created randomly or heuristically which represent the initial population. The population evolves by applying three basic operations: selection crossover and mutation with probability. For the Initial Population, the cryptanalysis process begins with randomly generated numbers between \( [0, 1] \) as the key size for n chromosomes and sorting these numbers in ascending order. The sequence of these numbers represents the candidate keys (chromosomes). Each chromosome represents the candidate key which it uses to decrypt the ciphertext and then calculate the fitness value to determine the best chromosome (candidate key).

For the Selection operator, selects chromosomes in the population for reproduction. The better chromosome has the opportunity to select more times to reproduce. Many selection procedures have been proposed, this paper used Roulette-wheel selection (that are described in section 2) to attack nonlinear stream cipher systems, which it used to selecting potentially useful solutions for reproduction. The chromosome (sequence) with high fitness has a higher probability of participate one or more offspring to the next generation. For the Crossover operator, two chromosomes are combining to produce a new generation that possesses both their characteristic. There are several crossover techniques, this thesis used single-point crossover (that are described in section 2) with probability of crossover \( p_c \) equal to 0.7 to attack streamcipher. For the Mutation operator, this process is used to maintain diversity in population from one generation to the next generation in order to obviate local minima. For this paper, a simple two point mutation is used. This process uses to select two individuals randomly on population in chromosome and swap between them. If the random number that generated to represent the candidate key is equal to 0.1.

For the Fitness Function calculation in this paper, equation (7) is used to calculate the fitness function of GA to attacks stream cipher. For the GA parameters there are a set of values which are considered as the most appropriate to attacks stream cipher by GA. Table (1) shows the different parameters of GA to cryptanalysis stream cipher Systems.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Length</td>
<td>KeyLen</td>
<td>[24]</td>
</tr>
<tr>
<td>Text Length</td>
<td>TxtLen</td>
<td>[10-100]</td>
</tr>
<tr>
<td>Number of chromosomes</td>
<td>Popsie</td>
<td>[20,100,200]</td>
</tr>
<tr>
<td>Maximum number of Iteration</td>
<td>MaxIter</td>
<td>[100-300]</td>
</tr>
<tr>
<td>Probability of crossover</td>
<td>P_c</td>
<td>0.7</td>
</tr>
<tr>
<td>Probability of mutation</td>
<td>P_m</td>
<td>0.1</td>
</tr>
</tbody>
</table>

4.3 Using PSO algorithm to cryptanalysis stream Cipher

In Evaluation For the initial population, the cryptanalysis process begins with randomly generated numbers between \([-1, 1]\) as the key size for n particles. The sequence of these numbers represents the candidate keys (particles). Randomly generates Velocity for each particle which it's bounded to some minimum and maximum values \([V_{max}, V_{min}]\) where \(V_{min}=-V_{max}\) and it uses to reinforces the local search reconnoitering of the problem space. Each particle represents the candidate key and use to decrypt the ciphertext and then calculate the fitness value to determine the best particle (key). For the evaluation process, the fitness value for each particle (candidate key) must be calculated for each generation.

Table (2) shows the most parameters of PSO that preferred to be used to decrypt stream cipher.

DOI: 10.9790/0661-1901047884 www.iosrjournals.org 81 | Page
Table (2): PSO parameters to attack stream cipher

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of particles in the swarm</td>
<td>Popsize</td>
<td>[20-200]</td>
</tr>
<tr>
<td>Number of Key</td>
<td>KeyLen</td>
<td>[12-24]</td>
</tr>
<tr>
<td>Length of text</td>
<td>TtxLen</td>
<td>[10-100]</td>
</tr>
<tr>
<td>The maximum number of iteration</td>
<td>MaxIter</td>
<td>[100-300]</td>
</tr>
<tr>
<td>The maximum of velocity</td>
<td>Vmax</td>
<td>4</td>
</tr>
<tr>
<td>The minimum of velocity</td>
<td>Vmin</td>
<td>-Vmax</td>
</tr>
<tr>
<td>Inertia Weight</td>
<td>W</td>
<td>[0.4-0.9]</td>
</tr>
<tr>
<td>Acceleration parameter</td>
<td>C1,C2</td>
<td>[0.5-2]</td>
</tr>
<tr>
<td>Random number between [0,1]</td>
<td>r1,r2</td>
<td>[0-1]</td>
</tr>
</tbody>
</table>

V. Comparison Results of cryptanalysis system between GA and PSO

The following Tables (3,4,5) shows the results of applying proposed cryptanalysis system GA and PSO For Popsize(20,100,200) and MaxIter(10,300) For TtxLen=100,40,10 characters, the following notations are used:
Popsie = Population size
MaxIter= Maximum Iteration
BF=Best Fitness
T/sec=Time/second
T.T/sec=Total Time/second
Iter_Num= Iteration_Number

Table (3) results of applying GA and PSO For Popsize(20,100,200) and MaxIter(100,300) For TtxLen=100 characters

<table>
<thead>
<tr>
<th>Popsize</th>
<th>MaxIter</th>
<th>GA</th>
<th>BF</th>
<th>T/sec</th>
<th>T.T/sec</th>
<th>Iter_Num</th>
<th>PSO</th>
<th>BF</th>
<th>T</th>
<th>T.T/sec</th>
<th>Iter_Num</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>100</td>
<td></td>
<td>0.5869</td>
<td>2.09</td>
<td>15.91</td>
<td>13</td>
<td>0.6188</td>
<td>23</td>
<td>16.31</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>100</td>
<td>0.5496</td>
<td>0.67</td>
<td>48.30</td>
<td>4</td>
<td>0.5520</td>
<td>0.58</td>
<td>47.20</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>300</td>
<td>0.5896</td>
<td>3.21</td>
<td>79.33</td>
<td>4</td>
<td>0.5656</td>
<td>1.21</td>
<td>82.32</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>100</td>
<td>0.6023</td>
<td>4.84</td>
<td>238.36</td>
<td>6</td>
<td>0.5556</td>
<td>0.82</td>
<td>240.98</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>300</td>
<td>0.0648</td>
<td>4.86</td>
<td>163.88</td>
<td>3</td>
<td>0.5850</td>
<td>3.29</td>
<td>160.47</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.5694</td>
<td>5.62</td>
<td>476.26</td>
<td>4</td>
<td>0.5760</td>
<td>1.58</td>
<td>460.32</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. (1) shows the comparison between GA and PSO in cryptanalysis system for Popsize (20) and MaxIter(100) for TtxLen=100

Figure (1) comparison between GA and PSO in cryptanalysis system for Popsize (20) and MaxIter(100) for TtxLen=100

DOI: 10.9790/0661-1901047884 www.iosrjournals.org 82 | Page
Table (4) shows the results of GA For Popsize(20,100,200) and MaxIter(100,300) For TxtLen=40 characters. **Table (4)** results of applying GA and PSO For Popsize(20,100,200) and MaxIter(100) For TxtLen=40 characters.

<table>
<thead>
<tr>
<th>Popsize</th>
<th>MaxIter</th>
<th>GA</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BF</td>
<td>T/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iter_ Num</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>0.5906</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.5938</td>
<td>0.21</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>0.6225</td>
<td>25.48</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.6262</td>
<td>20.30</td>
</tr>
<tr>
<td>200</td>
<td>100</td>
<td>0.6061</td>
<td>30.10</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.6023</td>
<td>5.60</td>
</tr>
</tbody>
</table>

Fig. (2) shows the comparison between GA and PSO in cryptanalysis system for Popsize (100) and MaxIter(100) for TxtLen=40

![Graph](image)

**Figure(2)** comparison between GA and PSO in cryptanalysis system for Popsize (100) and MaxIter(100) for TxtLen=40

Table (5) shows the results of GA For Popsize(20,100,200) and MaxIter(100,300) For TxtLen=10 characters. **Table (5)** results of applying GA and PSO For Popsize(20,100,200) and MaxIter(100,300) For TxtLen=10 characters

<table>
<thead>
<tr>
<th>Popsie</th>
<th>MaxIter</th>
<th>GA</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BF</td>
<td>T/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iter_ Num</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>0.6250</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.6750</td>
<td>0.83</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>0.6250</td>
<td>1.96</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.6625</td>
<td>27.25</td>
</tr>
<tr>
<td>200</td>
<td>100</td>
<td>0.6750</td>
<td>12.25</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.6625</td>
<td>13.92</td>
</tr>
</tbody>
</table>

Fig. (3) shows the comparison between GA and PSO in cryptanalysis system for Popsize (20) and MaxIter(100) for TxtLen=10

![Graph](image)

**Figure(3)** comparison between GA and PSO in cryptanalysis system for Popsize (20) and MaxIter(100) for TxtLen=10
VI. Conclusion

1) The cryptanalysis using GA and PSO can find the optimal solution for text with lengths of 10 characters as shown in Tables 3, 4 and 5. 
2) We conclude that the PSO cryptanalysis system performs better than GA in terms of time as shown in Tables 3, 4 and 5. 
3) As shown in Tables 3, 4 and 5the results of applying GA, PSO to cryptanalysis stream cipher, we notice that 5 iterations are enough to find the best solution for the PSO but this number of iterations are not enough for GA to find the best solution. 
4) As shown in Table(3) we conclude that the best results of GA and PSO inTxtLen=10 characters is Popsizes=20 and MaxIter=100. 
5) As shown in Table (4) the best results of GA and PSOinTxtLen=40 characters is Popsizes=100 and MaxIter=100. 
6) As shown in Table (5) we conclude that the best results of GA and PSO in TxlLen=10characters when Popsizes=20 and MaxIter=100. 
7) From a sequent 4,5 and 6, we conclude that Popsizes=20 and MaxIter=100 is enough to find the optimal key. 

Acknowledgements

I would like to thank my advisor Prof.Dr.Salim AL-Ageelee for motivation and support in presenting this paper.

References