DOI: http://dx.doi.org/10.21123/bsj.2020.17.2(SI).0682

Using Evolving Algorithms to Cryptanalysis Nonlinear Cryptosystems

Faez Hassan Ali [1] *Riyam Noori Jawad* [2*]

Received 7/9/2019, Accepted 23/4/2020, Published 23/6/2020

Abstract:

In this paper, new method have been investigated using evolving algorithms (EA's) to cryptanalysis one of the nonlinear stream cipher cryptosystems which depends on the Linear Feedback Shift Register (LFSR) unit by using cipher text-only attack. Genetic Algorithm (GA) and Ant Colony Optimization (ACO) which are used for attacking one of the nonlinear cryptosystems called "shrinking generator" using different lengths of cipher text and different lengths of combined LFSRs. GA and ACO proved their good performance in finding the initial values of the combined LFSRs. This work can be considered as a warning for a stream cipher designer to avoid the weak points, which may be found in the stream cipher, and may be explored by the cryptanalysts. This work can find the optimal solution for text with minimum lengths of 20 characters and 100 iteration were very enough to find the real initial values of key stream.

Key words: Ant Colony Optimization (ACO), Cryptanalysis, Genetic Algorithm (GA), Shrinking Generator, Stream Cipher.

Introduction:

The basic terminology is that cryptography term refers to the science of configure the ciphers; Cryptanalysis term is the science of extraction the clear text without knowing the key. It is a method of conversion ciphertext inside a plaintext without access to the key (1); while cryptology refers to the study of both. The Cryptographic algorithms are ordinarily divided into two parts, Secret algorithms and Public-key algorithms. Secret algorithms need to that communicating parties to share the same secret key. Symmetric key cipher can be sparely collected to Stream cipher and Block cipher. Stream ciphers encipher one byte or one bit at a time (2). For practical reasons, the bit-stream generator should be applied as an algorithmic procedure, so that the cryptographic bit stream can be produced by both users. The bit-stream generator is a key-controlled algorithm and must generate a bit stream that is cryptographically strong. Now, the two users require only sharing the generating key, and each can generate the key stream.

Pseudo-noise sequence created by the linear feedback shift registers (LFSRs) which utilized in cryptography field with some nonlinear combining functions have been suggested as running key generators in the stream cipher, test-pattern generation, signature analysis, uses in digital broadcasting and communications. T. Siegenthaler has stated that the number of attempts to attack these ciphers can be minimizing by using correlation ways (3). Genetic Algorithm (GA) depends on the soft computing ideas of normal operators (4). GA is a good candidate for the perfect cases to optimize and search problems (4). The algorithm has passed implementation on Maximum-Clique problem, Vertex-Cover problem, Regression testing, N-puzzle problem (5) and Traveling Salesman Problem (TSP) (6).

Abd et al attacked stream cipher systems using GA (7). Al-Ageelee and et al attacked Nonlinear Stream Cipher Cryptosystem (NLSCC) based on enhanced PSO (8).

Ant Colony Optimization (ACO) (9) is a well-known meta-heuristic that was successfully utilized to generate approximate solutions for a large variety of optimization problems.

ACO are used to attack Data Encryption Standard (DES) by Khan and et al (6). Also, Grari and et al (9) suggested Cryptanalysis of Simple Substitution Ciphers Using ACO. ACO used to attack Pointcheval's Identification Scheme by M.F.

[1] Mathematics of Department, College of Science, Mustansiryah University, Baghdad, Iraq.
Email: faezhassan@uomustansiriyah.edu.iq
ORCID ID: https://orcid.org/0000-0003-2371-069X
[2] Techniques of Quality Management Department, Technical College of Management, Middle Technical University, Baghdad, Iraq.
[*]Corresponding author: qweenr80@gmail.com
[*]ORCID ID: https://orcid.org/0000-0002-9031-8543

Uddin and et al (10), H. Grari and et al attacked Knapsack cipher (symmetric key) using ACO (11), S. Khan and W. Shahzad, F. Aslam Khan attacked four -Rounded DES using ACO when DES consist of 16 rounds (12).

In the paper of (6), he used a block cipher while in suggested work, a stream cipher have been used and in the paper (9) use a classical cipher (simple substitution cipher) but suggested work use a modern ciphers (stream ciphers), finally in the paper (12) use a block cipher for four rounds of DES while the suggested work use a stream cipher.

In this paper, ACO and GA as an EA's in new methods have been investigated as automated cryptanalysis system to attack one of the NLSCC called "shrinking generator" using different lengths of ciphertext and different lengths of combined LFSRs using the hardest type of attacking which is called "cipher text-only attack" depending on statistical properties of plaintext as a fitness function.

The rest of this paper is presented as follows: in addition to introduction, introduced Cryptosystems of Stream Cipher have been presented, then the EA which is represented by GA and ACO. Then the fully automated cryptanalysis system using evolutionary computation algorithm (GA and ACO) are presented, then the results and finally, conclusions and recommendations are presented.

The problem and aim of this paper:

The problem of this paper is to attack one of the nonlinear stream cipher generators using one of the evolving algorithms using ciphertext only attack.

In this paper and for the first try, the shrinking generator have been attacked as one of the stream cipher generators, using one of the new important evolving algorithms which is Ant Colony Optimization (ACO) method.

The objective of this attack to find the true initial values of the combined LFSRs of shrinking generator. This attack is applied for the first time, but of course may be there are other evolving algorithms that are used to attack (SG) or/and the ACO is used to attack other stream cipher generators.

Cryptosystems of Stream Ciphers

Stream ciphers form a significant category of symmetric-key enciphering schemes. They enciphered characters of individuals usually in binary digits of a clear-text message one at a time, using conversion of enciphering which it changes with time. One of the most important characteristics of stream cipher, with no error propagation characteristic, hence the transmission errors are highly probable (13). The attacks methods on stream ciphers can be categorized as; cipher texts-only attack, chosen clear text- attack, known clear-text attack and chosen ciphertext -attack. Also there are other kinds of attacking methods implemented upon stream cipher, several of these attacks are: Linear Consistency Attack, Determine and Guess attacks, Algebraic attacks and Inversion Attacks (14). In this paper ,the focus is on the *cipher text only attack*. This type known as difficult type of attacks for the cryptanalysis field no others data output about or the algorithm of cipher, but may contain information about the allocation of the clear text, like the language of the enciphered a message (15).

Many cryptosystems of stream cipher have been found. In this paper the interest has been in Shrinking Generator (SG) is a form of pseudorandom number generator intended to be used in a stream cipher.

Shrinking Generator (SG)

The SG utilizes a different form of clock control than the previous generators (such as gaffe and bruer generators). Take two LFSRs: R1 and R2. Clock both of them. If the output of R1 is 1, then the output of the generator is R2. If the output of R1 is 0, discard the two bits, clock both LFSRs, and try again. This idea is simple, reasonably efficient, and looks secure. Even so, it’s new (16). The following steps are repeated until key-stream of desired length is produced. The main steps of Shrinking Generator algorithm (SG) which are as follows in algorithm 1 (16):

Algorithm 1

```
Let Registers R1 and R2 are clocked by procedure Move.
INPUT: Length of Registers (R1, R2), with output (x1, x2) respectively, Initial setting and polynomial function, L (Length of needing bits).
i=0;
WHILE i <=L
    x1=MOVE(R1);
    x2=MOVE(R2);
   IF x1==1
     i=i+1;
    K(i)=x2;
  ENDIF
OUTPUT: K: Generated Key-stream
ENDWHILE
```

Evolving Algorithms (EA)

Evolutionary algorithms which model natural evolution processes have been successfully used for optimization. In this paper the focus has been on GA and ACO and utilize it in cryptanalysis field.

Ant Colony Optimization (ACO)

The basic idea of ACO was taken by the determine foods behavior of ant colonies that determine the shortest bath between ant's nest and a resource food by exchanging the data by pheromone deposited on the trips. This pheromone data is utilized to determine the directing of the search path and let ants collaborate with each other in order to determine high quality of good solutions in a large search space (17). The main Pseudo-code and formula of ACO are as follows in algorithm 2:

Algorithm 2

```
Procedure ACO_Metaheuristic
WHILE (not_termination)
    generateSolutions()
    pheromoneUpdate()
    end while
END procedure
```

Genetic Algorithm (GA)

GA's are optimization search algorithms which depend upon the idea of natural selection and natural genetics (18).

An abstract algorithm 3 of the GA is as the following:

Algorithm 3

```
Gen=0;
InitializeProcess Ge(P);
Evolve Ge(P);
WHILE (GA conditions has not Stopped)
Generation = Gen + 1;
SelectProcess Ge(P) from Ge(P-1);
CrossoverProcess Ge(P);
MutateProcess Ge(P);
Evolve Ge(P);
END (While)
Stop the GA.
```

GA-ACOCS Using EA

In this section a new cryptanalysis system for stream cipher have been suggested, that depends on GA and ACO algorithms. This cryptanalysis system is called **G**enetic **A**lgorithm- **A**nt **C**olony **O**ptimization **C**ryptanalysis **S**ystem (GA-ACOCS).

The comparison was between GA and ACO when these methods are suggested in this work.

Before applying the GA-ACOCS, the encryption system using SG must be discussed.

Let's have plaintext with Length L bits to obtain Cipher text (C). The Encryption System using SG (ESSG) steps are as follows in algorithm 4:

Algorithm 4

```
INPUT: Plaintext (P) characters with length M bytes (ASCII Codes).
Convert the plaintext to L=8*M bits.
FOR i=1:L
    $K_i$=CALL $SG_i$ ;
    $C_i$= $K_i$ xor $P_i$;
ENDFOR(i)
OUTPUT: Ci
END.
```

For any cryptanalysis system using EA, the following main stages must be available:

Step(1) (*Initialization step*): in the Initial generation, the attacking process starts with random generation process for values in range coding of {0, 1} as the key-stream size for n individuals. The sequence of the generated values indicates the desired keys (individual). Each individual indicates to the candidate key which will be utilized for decrypting the ciphertext and then computing to the fitness function value to save the optimal fitness (the right key).

Step(2) (*Fitness Function Calculation*): This stage will apply each iteration which includes calculate the fitting value. The fitting value is represented by taking the best-fits of the statistical properties of plaintext depending on the used language. In English language in any plain message, the "0" percentage is about 60% of whole text. So the best fitness must be ≥ 0.60; this rate was as a measure factor in this work, which has been changed depending on various clear-text size. In this paper the fitness function algorithm for ACO and GA in GA-ACOCS can be stated as follows in algorithm 5:

Algorithm 5

```
Function Fit=F-Fitness (L,P)
  S=0;
  FOR i=1:L
        IF P(i)==0
           S=S+1;
        ENDIF
   ENDFOR
   Fitness= S/L;
END
```

Step(3) (*Update Population*): This stage is related to used EA (ACO and GA), which means that the update stage in ACO algorithm is represented in pheromone update stage, while in GA it has been represented in three basic operators; selection, crossover and mutation.

In this paper, three samples of plaintext (T1, T2 and T3) have been suggested with sizes 100, 50 and 20 characters, respectively. These three samples give different fitness values for each sample. Table 1 shows the optimal fitness values for each sample. These value are calculated as mentioned above in algorithm 5.

Table 1. Optimal fitness values for different plaintext sizes.

Sample	Size/ Binary	Fitness Value
T1	100*8=800 bits	0.6342
T2	50*8=400 bits	0.6272
T3	20*8=160 bits	0.6062

Many kinds of SG cryptosystem have been tried, but only two kinds (SG1 and SG2)have been focused with LFSR lengths 5 and 7 for SG1 and 7 and 9 for SG2. Table 2 presents the main parameters of GA-ACOCS.

Table 2. EA Parameters used in GA-ACOCS.

Symbol	Description	Value
MaxIter	Maximum Iteration	50-100
P	Population size (Number of Individuals or ants)	20-50
N	Parameter which represent sum of R_1 and R_2	12-14
L	Cipher text Length	[20-200]

ACO in GA-ACOCS

ACO has been successfully implemented on various problems (applications) in the field of search and optimization. It is a recursive protocol that contains ant size of tours. These tours are created randomly which correspond to the initial generation. The population has been evolved by applying three basic processes: initialization, fitness calculation, pheromones updating and evaporation process. Table 3 states the ACO parameters were used in GA-ACOCS.

Table 3. ACO Parameters used in GA-ACOCS.

Symbol	Description	Value
Q	Parameter which will be used in pheromone updating stage	1
τ	Initial pheromone	0.5
α	Pheromone exponential weight	0.3
ρ	Evaporation rate	0.1

GA in GA-ACOCS

For the ***Initial Generation***, in the GA-ACOCS system starts with randomly generated values in range {0, 1} that are considered as the key size for n chromosomes according to the lengths of LFSRs and sorting these numbers in ascending order. The sequence of these values indicates the correct key (individuals).

While in the ***Selection operator***, chose chromosome in the population for crossover process which is considered to be from GA operators. Roulette-wheel selection is the method which will be utilized in the good solutions selection more often than the other solutions.

In the ***Crossover Process***, two individuals have been selected to crossover process completion to generate a new population. In this work, a single-point crossover method has been used. For ***Mutation operator***, flip mutation method has been used, which changed 1 to 0 and 0 to 1.

In the Fitness Function, the algorithm mentioned in step (2) of section 4 have been used. In the GA variables which represent the mostly used method to cryptanalysis stream cipher crypto-systems by GA which are shown in Table 4. The basic algorithm of cryptanalysis system stated in GA-ACO Cryptanalysis System in algorithm 6

Table 4. GA Parameters Configuration in GA-ACOCS.

Symbols	Description	Value
C_p	Crossover-of- Probability rate	0.5
M_p	Mutation- Probability rate	0.2

Algorithm 6

GA-ACOCS Algorithm

```
Step(1): INPUT: lengths $R_1$ and $R_2$, C.
Step(2): specify EA = ACO (1) / GA(2).
Step(3): CALL initialization function.
Step(4): FOR k = 1:MaxIter
        CALL Fitness function.
        For each individual: FIND Best-Fitness.
        CALL Update function of the population according to the used EA New-Pop=UPDATE (EA).
ENDFOR {k}.
Step(5): OUTPUT: Best_Fitness + Best Individual (Optimal Key).
Step(6): Find Plain=Decryption (using SG).
Step(8): END.
```

Results of GA-ACOCS Using EA (GA and ACO)

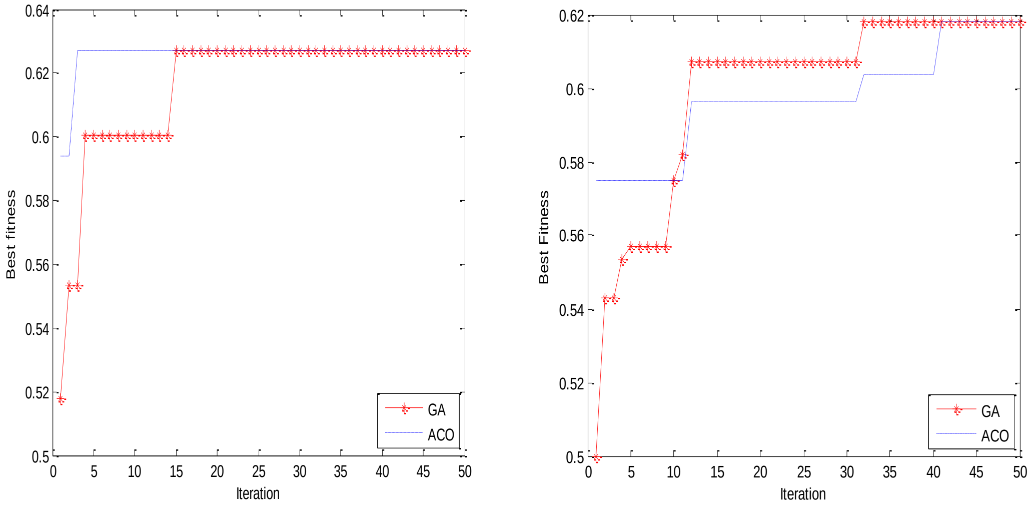
Before discussion of the results of cryptanalysis of SG1 and SG2 using GA-ACOCS in Tables 5 and 6, some notations must be listed:

Popsize	:	Population size.
BF	:	Best Fitness.
T/sec	:	Time in Seconds.
Iter_Num	:	Iterations Number.
CT	:	Ciphertext.

Table 5. Applying GA and ACO with Popsize=50 for SG1.

Length of	GA			ACO		
CT (Bytes)	BF	Iter-Num	T/sec	BF	Iter-Num	T/sec
T1	0.6342	93	21.42	0.6342	61	19.32
T2	0.6272	15	1.01	0.6272	3	0.01
T3	0.6062	40	2.01	0.6062	10	1.02
Average		49	8		25	6

Figure 1 shows the developing of fitness values for GA and ACO (a): for T1 for SG1 and (b): for T2 for SG1.



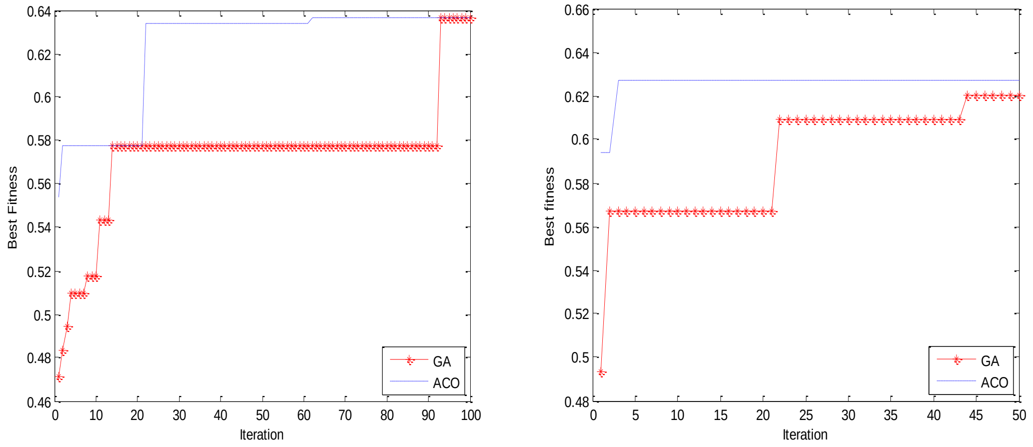
(a) **(b)**

Figure 1. The developing of fitness values for GA and ACO (a): for T1 for SG1 and (b): for T2 for SG1.

Table 6. Applying GA and ACO for Popsize=50 for SG2.

Length of	GA			ACO		
CT (Bytes)	BF	Iter-Num	T/Sec	BF	Iter-Num	T/Sec
T1	0.6342	80	62.42	0.6342	70	50.32
T2	0.6272	35	20.01	0.6272	20	10.01
T3	0.6062	43	42.01	0.6062	23	12.02
Average		52	41.44		37	24.15

Figure 2 shows the developing of fitness values for GA and ACO (a): for T2 for SG2 and (b): for T3 for SG2.



(a) **(b)**

Figure 2. The developing of fitness values for GA and ACO (a): for T2 for SG2 and (b): for T3 for SG2.

Conclusions and Recommendations:

1. The ciphertext length has serious effects on GA-ACOCS results for the stream ciphers, the obtained results are more precise whenever the length of ciphertext as high as possible.
2. As the stream cipher cryptosystem be complicated as the needed number of generations and the consuming time are increased to find the actual initial key.
3. From tables 5 and 6, the analytical study of applying GA and ACO give the following results:
4. The GA-ACOCS can find the optimal solution for text with minimum lengths 20 characters.
5. The applying of ACO performs better than GA in term of CPU time have been concluded.
6. From applying GA-ACOCS, 100 iterations are very enough to find the real initial values of SG1 and SG2 have been proved.
7. As a future work, GA-ACOCS can be used to attack more complicated cryptosystems in the field of stream cipher (Geffe, or Bruer with three or more LFSRs).
8. As a future work, GA-ACOCS can be extended to break the other modern cipher systems like Block Cipher and public key.
9. As a future work to improve the performance of GA and ACO, a hybrid between other soft computing algorithms (e.g. simulated annealing or decent algorithm), may be suggested.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Middle Technical University.

Reference:

1. Behrouz AF. Data Communication and Networking. Fourth Edition; 2007.
2. Joseph L. Cryptanalysis and Design of Synchronous Stream Ciphers. U.D.C., 2006.
3. Siegenthaler T. Decrypting a class of stream ciphers using ciphertext only. IEEE comput. 1985 Jan 1(1):81-5.
4. Salim A, Riyam N. Cryptanalysis of Stream Cipher Cryptosystem Based on Soft Computing Techniques. IOSR-JCE; 2017 19 (1): 78-84.
5. Harsh B, Neha S. Genetic based algorithm for N-Puzzle problem. IJCA؛ 2012; 51(22).
6. Yen L , Dun Y, Chieh C. Evolutionary algorithm to Traveling Salesman Problems. COMPUT MATH APPL; 2012; 64(5): 788-797.
7. Ali A, Hameed A, Wasan S. Attacking of stream Cipher Systems Using a Genetic Algorithm. University of Thi-Qar Journal; 2013; 8(3):188-194.
8. Salim A, Riyam N. Cryptanalysis of Nonlinear Stream Cipher Cryptosystem based on Improved Particle Swarm Optimization, International Journal of Applied Information Systems (IJAIS); 2017; 11(11).
9. Hicham G, Ahmed A, Khalid Z, Mohamed B, Jaafar G. Cryptanalysis of Knapsack Cipher Using Ant Colony Optimization, the second international conference on smart applications and data analysis for smart cites, 2018.
10. Mohammed F, Amr M .Cryptanalysis of Pointcheval's Identification Scheme Using Ant Colony Optimization, IEEE, 2007.
11. Grari H, Azouaoui A, Zine-Dine K. Ant Colony Optimization for Cryptanalysis of Simplified-DES. InInternational Conference on Advanced Intelligent Systems for Sustainable Development 2018 Jul 12 (pp. 111-121). Springer, Cham.
12. Salabat K, Waseem S, Farrukh A. Cryptanalysis of Four-Rounded DES using Ant Colony Optimization, IEEE, 2010.
13. Azhar M. Mathematical Analysis of Design Parameters for Nonlinear Stream Cipher Systems, M. Sc. Thesis, University of Technology, 2005.
14. Christian R. Side Channel Analysis of Stream Ciphers, Austria, 2004.
15. Tariq S, Faez H. Modification of Some Solution Techniques of Combinatorial Optimization Problems to Analyze the Transposition Cipher. Mathematical Theory and Modeling, IISTE, 2014.
16. Schneier B. Applied Cryptography, Second Edition: Protocols, Algorithms and Source Code in C, 1996.
17. Bijaya K, Swagatam D, Ponnuthurai NS, Subhransu SD. Swarm, Evolutionary and Memetic Computing. Springer-Verlag Berlin Heidelberg, 2010.
18. Goldberg DE. Genetic Algorithms in search, Optimization, and Machine Learning, 1989.

استخدام الخوارزميات التطورية لتحليل انظمة التشفير غير الخطية

فائز حسن علي [1] **ريام نوري جواد[2]**

[1] قسم الرياضيات، كلية العلوم، الجامعة المستنصرية، بغداد، العراق.
[2] قسم تقنيات ادارة الجودة الشاملة، الكلية التقنية الأدارية /بغداد، الجامعة التقنية الوسطى، بغداد، العراق.

الخلاصة:

في هذا البحث، نتحرى عن استخدام الخوارزميات التطورية (EA's) لتحليل أحد أنظمة التشفير غير الخطية التي تعتمد على وحدة السجلات الزاحفة لتبادل البيانات الخطية (LFSR) باستخدام طريقة هجوم النص المشفر فقط. الخوارزمية الجينية (GA) و خوارزمية خلية النمل (Ant Colony Optimization (ACO)) التي استخدمت في مهاجمة أحد أنظمة التشفير غير الخطية المسماة " Shrinking Generator" باستخدام أطوال مختلفة من النص المشفر وأطوال مختلفة من LFSRs المدمجة أثبتت أدائها الجيد في إيجاد القيم الأولية لل LFSRs المدمجة.

الكلمات المفتاحية : تحليل الشفرة، الشفرة الأنسيابية، خوارزمية خلية النمل، الخوارزمية الجينية، مولد Shrinking.