International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 5, October 2019, pp. 4287~4295
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i5.pp4287-4295

A cryptanalytic attack of simplified-AES using ant colony optimization

Hicham Grari, Ahmed Azouaoui, Khalid Zine-Dine
LAROSERI Lab., Faculty of Sciences, Chouaib Doukkali University, Morocco

Article Info

Article history:

Received Feb 23, 2019
Revised Apr 22, 2019
Accepted Apr 30, 2019

Keywords:

ACO
Cryptanalysis
Meta-heuristic
Pheromone
S-AES

ABSTRACT

Ant colony Optimization is a nature-inspired meta-heuristic optimization algorithm that gained a great interest in resolution of combinatorial and numerical optimization problems in many science and engineering domains. The aim of this work was to investigate the use of Ant Colony Optimization in cryptanalysis of Simplified Advanced Encryption Standard (S-AES), using a known plaintext attack. We have defined the essential components of our algorithm such as heuristic value, fitness function and the strategy to update pheromone trails. It is shown from the experimental results that our proposed algorithm allow us to break S-AES cryptosystem after exploring a minimum search space when compared with others techniques and requiring only two plaintext-ciphertext pairs.

Corresponding Author:

Khalid Zine-Dine,
LAROSERI Lab., Faculty of Sciences,
Chouaib Doukkali University,
Route Ben Maachou, 24 000, El Jadida, Morocco.
Email: zinedine@ucd.ac.ma

1. INTRODUCTION

Cryptology is one of the most significant techniques for achieving information security which is become a vital need in communication networks and computer systems. Cryptology is the science of building and analysing different encryption and decryption systems. It consists of two subfields; Cryptography and Cryptanalysis. Cryptography is the study of building new powerful and efficient encryption and decryption algorithms. Cryptanalysis is the art of deciphering communications that are secured by cryptography, it is used to find weakness and flaws of ciphers.

Actually, research in the cryptology field are increasingly using evolutionary techniques, encouraged by the promising obtained results. Specially, Ant Colony Optimization (ACO) which is a well-known meta-heuristic that was successfully used for solving a various real-world optimization problems. Recently , Ant Colony Optimization was used to attack DES (Data Encryption Standard) by Salabat Khan et al. in [1]. Also, Grari et al. [2] proposed a novel attack of Simple Substitution Ciphers based on Ant Colony Optimization.

In this paper, we introduce a novel cryptanalytic attack of Simplified Advanced Encryption Standard (S-AES) using Ant Colonny Optimization . we have modelled the cryptanalysis's problem to a combinatorial problem in order to apply ACO metaheuristic to break the Simplified-AES cryptosystem. We will show that our approach is significantly faster and requires a smaller number of plaintext-ciphertext pairs, when compared to others attacks.

The cryptanalysis of S-AES has been the subject of several previous works. Mainly, Musa et al. [3] attacked S-AES for the first time using linear and differential cryptanalysis, concluding that their linear cryptanalytic attack seems very attractive compared to a pure brute force attack, requiring 109 plaintext and the corresponding ciphertext pairs to attack only the first round in S-AES. Mansoori et al. [4, 5] applied

a linear cryptanalysis to S-AES . It has been shown that at least 116 plaintext and corresponding cipher text pairs are required to break the first round and 548 to break the second round, concluding that S-AES is vulnerable against linear attack. While, Simmons [6] performed an algebraic cryptanalysis to S-AES, by resolving a system of polynomial equation. However, some others attacks based on metaheuristics were carried out such as Valarmathi and Vimalathithan [7, 8] which attacked S-AES using Genetic Algorithm [7] and Particle Swarm Optimization [8], they break the key used in S-AES in a minimum search space compared to brute force attack. Rania Saeed and Ashraf Bhery [9] proposed cryptanalysis of S-AES using Intelligent Agent needing only one plaintext.

The remainder of this paper is organized as follows. In the next section, we introduce Simplified Advanced Encryption Standard. In section 3, we describe the Ant colony optimization meta-heuristic. The fully automated attack is given in section 7, with experimental results in section 5. Finally, conclusions are given in section 6.

2. SIMPLIFIED ADVANCED ENCRYPTION STANDARD (S-AES)

Simplified Advanced Encryption Standard (S-AES) is a block Cipher algorithm that takes a *16-bit* plaintext and *16-bit* key as input to generates *16-bit* ciphertext as output. The *16-bit* input plaintext are treated as *4x4* matrix of nibbles (a nibble is a *4-bits* block), called state. Each round takes a state and generates a new one to be used in the next round. Four transformations are used in S-AES: Substitution (SubNibbles), shift row, mix columns and add round key. Figure 1 shows encryption algorithm, key generation and decryption algorithm for S-AES.

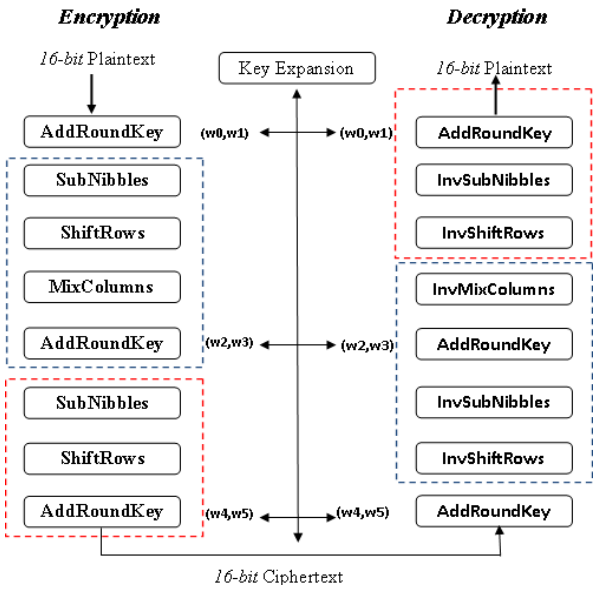

Figure 1. S-AES Encryption and decryption

SubNibbles (S-box): is the only nonlinear component in the algorithm that provides confusion effect, the substitution table used for this purpose is described in (1). It takes a *4-bit* input and produces a *4-bit* output. In the input nibble, the left 2 bits determine the row and the right 2 bits determine the column of the substitution table. The hexadecimal value at the junction of the row and the column is the output nibble.

$$S = \begin{bmatrix} 9 & 4 & A & B \\ D & 1 & 8 & 5 \\ 6 & 2 & 0 & 3 \\ C & E & F & 7 \end{bmatrix} \quad (1)$$

ShiftRows: In this step the, the first row of the state matrix remains unchanged. While the second row, a one-nibble circular shift is performed.

MixColumns: As a third step, the MixColumns transformation operates at the column of the matrix; each column of the state is transformed into a new column. The transformation is actually the matrix multiplication of a state column by a constant square matrix. The nibbles in the state column and constants matrix are interpreted as polynomials with coefficients in Galois Field GF (2). Multiplication of bytes is done in GF (2^4) with modulus the irreducible polynomial (x^4+x+1) to ensure that the result is still within the filed GF (2^4).

Let (a_0,a_1,a_2,a_3) the input nibbles of the MixColumns operation, the block at the output (b_0,b_1,b_2,b_3) is defined as follows:

$$\begin{bmatrix} b_0 & b_2 \\ b_1 & b_3 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} a_0 & a_2 \\ a_1 & a_3 \end{bmatrix}$$

AddRoundKey: The last stage of each round is to add the round key. It consists of the bitwise XOR of the 16-bit state matrix and the 16-bit Key round.

S-AES Key Expansion: In this process, three round keys are generated from the originated key, each key is used in different specific round, allowing increasing the security of S-AES. The keys used for encryption algorithm are also used for decryption.

The key expansion algorithm creates round keys word by word, where a word is an array of 2 nibbles, the algorithm produces 6 words, which are called

$W_0, W_1 \ldots W_5$.

Where: $K_0=W_0W_1$, $K_1= W_2W_3$ *and* $K2 = W_4W_5$.

From the original key we generate the two byte W_0 and W_1. Next ,we can produce the others words using algorithm described as follow :

S-AES Key Expansion
For $2 \leq i \leq 5$ *do*
  If $i \equiv 0 (mod\ 2)$ *then*
  $W_i = W_{i-2} \oplus RCON(i/2) \oplus SubNib(RotNib(W_{i-1}))$
  Else
  $W_i = W_{i-1} \oplus W_{i-2}$
  End if
End for

In this algorithm, *RCON*[i] = *RC*[i]0000, where RC[i] is defined as RC[i] =$x^{i+2} \in GF(2^4)$ so RC[1] = x^3 = 1000 and RC[2] = x^4 = x +1 = 0011. If N_0 and N_1 are two nibbles and their concatenation denoted as N_0N_1. The *RotNib an* SubNib functions are defined to be: $RotNib(N_0N_1)=N_1N_0$ and $SubNib(N_0N_1)=Sbox(N_0)Sbox(N_1)$ which means respectively rotate nibble and substitute nibble.

Decryption: Decryption is the reverse process of encryption. It takes a *16-bit* ciphertext, the *16-bit* key, and generates the original *16-bit* Plaintext. Similarly to encryption, decryption uses one pre-round and two round transformations, as shown in Figure 1. The processes performed during decryption are the inverse of those employed in encryption.

3. ANT COLONY OPTIMIZATION (ACO)

Optimization metaheuristics have a significant importance in determining efficient solutions of different complex and hard problems. Especially, Ant Colony Optimization, which represents a class of nature-inspired meta-heuristics, based on the behavior of real ant within their colonies. Dorigo et al. [10] proposes the first ACO algorithms in the early 1990s. The study of the behavior of ants within colonies inspires the development of these algorithms. However, ants are a social insects and their behavior is governed by the goal of colony survival being focused on the survival of individuals. First, ants explore randomly the area arounding their nest to search the food. When an ant finds food, it walks back to the colony leaving behind a pheromone trail on the ground that may depend on the quantity and the quality of the food. This pheromones trail will guide other ants to the food source. Old paths are less likely to be used because of the pheromone evaporation mechanism. This food supplies behavior has inspired the development of ACO, in particular, this ability to explore paths between food sources and their nest and finding the shortest one.

The first ACO algorithm, called Ant System (AS) introduced by Dorigo [10] was applied to Travelling Salesmen Problem. Other ACO variants mostly differ in the rule used for the solution construction and the pheromone update, including Ant Colony System (ACS) presented by Dorigo and Gambardella [11], and Min Max Ant System (MMAS) given by Stutzle and Hoos [12].

4. THE PROPOSED APPROACH

Assuming that we know a part of plain text P and his corresponding ciphertext C during the optimization process. Let z the number of known plaintext-ciphertext pairs P_i/C_i. The main goal is to find the key that allows to performing this encryption. Using our proposed algorithm, we generate a candidate key K_c, which is used to decrypt the cypher text C_i. Then, a candidate plaintext PG_i is generated. Furthermore, this candidate Plaintext is used in the fitness function defined in (3) to evaluate K_c. The Full Layout of our attack algorithm is described in Figure 2.

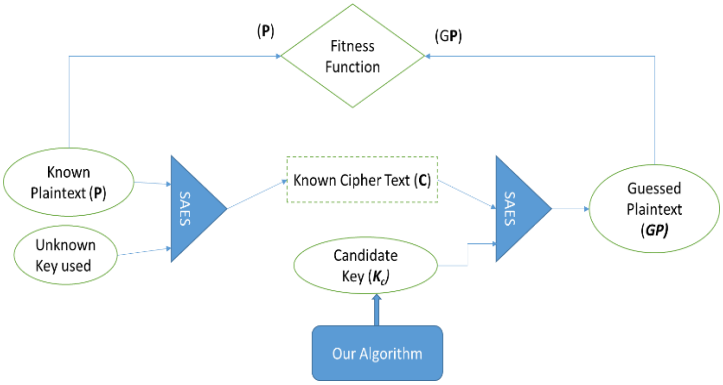

Figure 2. Layout of our attack algorithm

In order to apply the ACO metaheuristic, we have modelled the search space to a *n+1-nodes* graph (*n* represent the key length used by S-AES). Every node in the graph is connected to the next node by two different edges, the upper edge is equal to 0 and the lower edge is equal to 1 as shown in Figure 3.

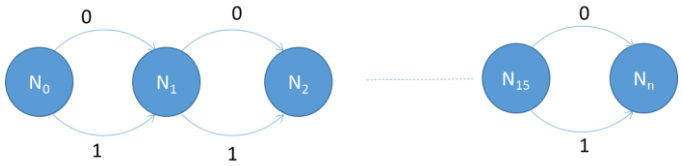

Figure 3. Search space for cryptanalysis of SAES

Each ant starts it tour from the start node N_0 moving from left to right; its tour is finished at the last node N_n. At each Node, an ant can only select a single edge during a particular tour. Therefore, each complete path from the node N_0 to N_{n+1} that comply with the precedence constraints is considered as a feasible solution , it will consist of *n-bit* binary string which is considered as a candidate key. After all the ants have completed their tours, all the solutions generated during the current Cycle are evaluated and compared by the objective function, the candidate key with the best fitness value in each Cycle is saved as a global best ant K_{Best}.

4.1. Solution construction

Each ant constructs a key using the function Probabilistic Stepwise Construction based on a probabilistic move of ants across the nodes. For an ant k , the probability P_{ij} to moves from a node *i* to node *i+1* following the path j is defined by (2).

$$P_{ij}(k) = \begin{cases} 0 & \text{if not allowed} \\ \frac{\tau_{ij}(k)^{\alpha}\, \rho_{ij}(k)^{\beta}}{\sum_{l \epsilon S} \tau_{il}(k)^{\alpha} \rho_{il}(k)^{\beta}} & \text{Otherwise} \end{cases} \quad (2)$$

Where S is the set of the possible path (0 and 1).

This probability of moving from a node to node depend on two parameters. The first, the pheromone trail τ(i,j) on the edge between the two nodes. And second the heuristic value ρ(i,j) representing the a priori knowledge of desirability of the choice. The parameters α and β are parameters which determine the relative influence of pheromone trail versus heuristic value.

4.2. Heuristic value

Heuristic value is essential for the generation of high-quality solutions in the early stages of the search process. Especially, when using heuristic value calculation method from the problem domain. In our approach, we have used a dynamic heuristic that has to be computed at each step of an ant's walk.

First, as explained previously, the candidate key with the best fitness value in each Cycle is saved as a global best ant (K_{Best}). Then, at every node, ants uses heuristic value calculated as shown in Figure 4.

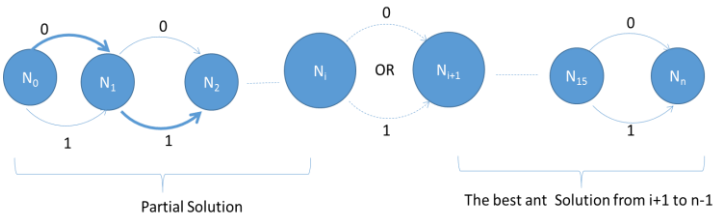

Figure 4. Heuristic value calculation

Let N_i the current node as shown in Figure 4, an ant has to decide which path to follow to move to the next node N_{i+1}. The j is the edge to be choose , only two values are possible i.e. either *0* or *1*. we define Hj as a concatenation of the current key (the partial solution) from 0 to i and the j value and K_{Best} *from i+2* to *n-1*. \

H_j= {*CurrentKey* (from *0* to *i-1*) | j | K_{Best} (i+1 to n-1)}

So, the concatenated binary string H_j represente a guessed key which is evaluated using the fitness function, the obtained value is used as a heuristic value in (2).

4.3. Fitness function

The quality of a generated key is assessed by its fitness function value. The main goal of a fitness function is to provide a measurable value for a given key that indicate its proxomity to the real key. It is a primordial component guiding the search algorithm to the best solutions within a large search space. Allowing so that to explore the search space more efficiently.

Let z the number of known plaintext-ciphertext pairs P_i / C_i. To evaluate a candidate Key K_c, a candidate plaintext GP_i is produced based on K_c. The fitness function F of the key K_c is defined in 3:

$$F(K_c) = \frac{\sum_{i=1}^{z} \#(\mathrm{GPi} \oplus Pi)}{z*16} \quad (3)$$

$\oplus$ represents the Xor operation, and # denotes the number of zeros in ($GP_i \oplus P_i$).

The range of F is *[0,1]*. Particularly, F is equal to 1, if all bits in GP_i are identical to P_i for any i in [1-z]. In this case, the generated key is the real one, so our goal is to maximize the fitness function.

4.4. Pheromone update

Initially, the quantity of pheromone in each edge is equal to τ_0 (initial pheromone value). Only the best solution (K_{best})in a particular Cycle (C) is allowed to update the pheromone values on the edges constituting the tour. The best ant information is also updated after each Cycle. The pheromones over the

edges constituting the tour of the best ant is updated using (4), so larger the fitness value, the greater is the amount pheromone concentrated.

$$\tau_{i,j} = \sigma * \tau_{i,j} + Q * F(\mathrm{K}) \quad (4)$$

Where Q is some constant and σ represent a pheromone evaporation (will be between 0 and 1), it consist to decreases the pheromone trail uniformly in all edges. This operation allows avoiding a premature convergence of the algorithm into a local optimal solution.

Outline of ACO Algorithm:

Step1: Perform initialization of pheromone

Step2: Repeat the followings steps

1. Complete the tours of (N) ants by making the decisions using probability (2)
2. Calculate fitness value for the tours of (N) ants according to (3)
3. Update best ant information and pheromone values on edges constituting the tours using (4)

Step3: If a maximum number of Cycle (C) have been attained or threshold of fitness function is reached then stop

Step4: Else go to ***Step 2***

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

Generally, the performances of any evolutionary computation algorithm are closely related to the parameters values. Therefore, one of the main challenge is to find the optimal parameters setting allowing to improve efficiency of our algorithm. The values of parameters assumed in this paper such as α, β (weight of pheromone and heuristic value), N (Number of Ants), C (Number of Cycle), 'Q' and σ were fine-tuned by a combination of several experiments in order to optimize the cryptanalysis process. The default value of the parameters was α=1, β=1, Q=2, σ=0.97 and $\tau_0 = 5$. We have implemented our algorithm with C++ language.

5.1. Key space analysis

The first part of the experiments is to determine the optimal number of ants to use that allows finding the key in a minimal search space. The experimental results obtained in this part are illustrated in Table 1. In This table, we can see the number of keys searched before locating the reel key and the average number of Cycle C (The average is calculated on 1000 launch) needed for each value of N.

Table 1. Experimental results for different values of N

Used Key	Number of cycles (C) needed	Number of Ants (N) used	Number of Keys browsed	Key found
C3F0	-	10	-	BE56
C3F0	-	20	-	C5F0
C3F0	-	30	-	C12A
C3F0	163	40	6520	C3F0
C3F0	122	50	6100	C3F0
C3F0	106	60	6360	C3F0
C3F0	92	70	6440	C3F0
C3F0	77	80	6160	C3F0
C3F0	66	90	5940	C3F0
C3F0	59	100	5900	C3F0
C3F0	50	110	5500	C3F0
C3F0	43	120	5160	C3F0
C3F0	42	130	5460	C3F0
C3F0	42	140	5880	C3F0
C3F0	41	150	6150	C3F0

As shown in Table 1, with a small number of ants (N<40), our algorithm cannot locate the real key, the number of ants is not enough large, thus the best solution found on each Cycle (which is used in pheromone update) is not enough good, which penalizes the convergence of the optimisation process to the right solution.

As noted in the Table 1, with the values N=120 we need 43 Cycles to locate the real key (C=43), the real key is found in a minimum search space. The maximum fitness value is reached after checking 5160 keys, thus being considerably lower (by a factor of 12) than the brute-force search space size (which is equal

to 2^{16} possible keys), and lower than GA[7] and PSO[8] which need to explore 6000 and 6905 keys, respectively , to locate the real one.

Figure 5 shows the evolution of the number of keys browsed before locating the real key under the number of ants used (N), It is clear from Figure that when the number of Ants is greater than 128, the search space increases rapidly with no improvement of the number of Cycle (C).

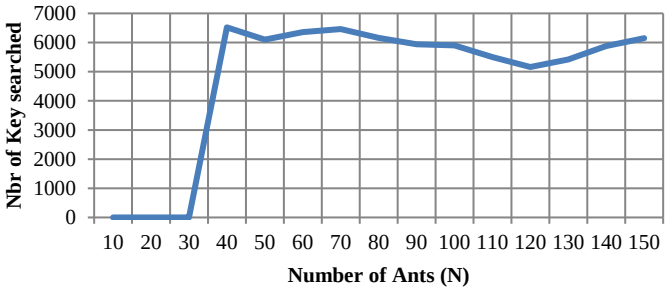

Figure 5. Number of key searched evolution

5.2. Relevance of the fitness function

The design of fitness function is a crucial step in evolution algorithms. In order to assess the relevance of our fitness function, the correlation coefficient 'Corr' between the cost function (3) and the number of corrected key elements (Number of correct bits in the key) is calculated using (5), for different values of z (number of known plaintext-ciphertext pairs used).

This coefficient is used to evaluate the relationship between the cost function and the number of key elements correctly recovered. In the case of a perfect direct (increasing) linear relationship we have Corr = 1, its means that we have a good mechanism for evaluation of generated key, therefore the convergence of our algorithm to the best key is most guaranteed. Inversely, in a perfect decreasing linear relationship we have Corr = -1. Figure 6 shows the value of the correlation coefficient between our fitness function and the number of corrected key elements, for different number of known plaintext-ciphertext pairs. To calculate this coefficient, we have used 1000 keys as sample, by calculating the fitness and the correct number of bits for each one.

$$Corr(x,y) = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (5)$$

Where x and y are two arrays of n elements.

As shown in Figure 6, the correlation coefficient increases with the number of pair used, but from 3 pairs required, we note a stagnation of this coefficient. The use of 3 pairs in the fitness function gives almost the same results as z=2, while the use of a single pair can drive the algorithm to a solution with a maximum fitness function value without being the right key.

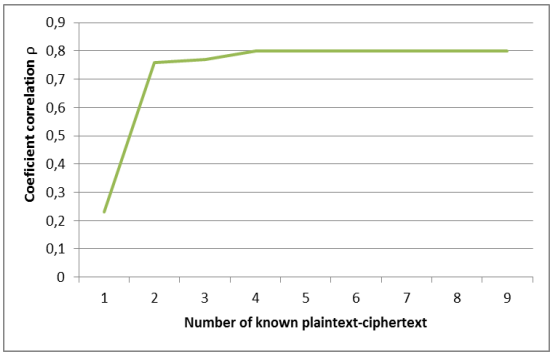

Figure 6. Correlation coefficient evolution through number of known plaintext-ciphertext

5.3. Number of plaintext-ciphertext pairs analysis

Our proposed approach allows us to find the key using only two known plaintext-ciphertext pairs, whereas in other attacks the number of plaintext-ciphertext pairs required is reported in Table 2.

Table 2. Number of plaintext-ciphertext required for attacking

Methodology	Rounds attacked	Number of Plaintext-Ciphertext pair required
Linear cryptanalysis Musa [3]	Round 1	109
Linear cryptanalysis Davood [5]	Round 1	116
	Round 1 & Round 2	548
Linear cryptanalysis Bizaki [4]	Round 1 & Round 2	96
Using GA Vimalathithan [7]	Round 1 & Round 2	3
Our Approch using ACO	Round 1 & Round 2	2

A fitness function based on only one pair of known plaintext-ciphertext (z=1 in the (3)), we can find a solution that maximizes the fitness function but without being the reel key. Therefore, the need for a second pair proves necessary to confirm that is the right key.

5.4. Pheromone sensitivity analysis

In order to analyse the pheromone trails effect within the solutions construction process. The number of ants (N) is fixed at 120 and β at 1. First, we used a rather strong pheromone concentration, illustrated by the value α = 2. As shown in Figure 7, we can observe as an early stagnation of research because of the algorithm has ceased exploring new possibilities. Inversely, with a too weak guidance of the search, we noticed an excessive exploration of the search space, this undesirable behaviour is illustrated in the Figure8 with α is equal to 0.5. As a result, the search algorithm is not able to converge to an optimal solution.

The best results are obtained with α = 1.2. Thus, a quick convergence of the algorithm to the correct key is observed in this case. Concluding that the right parameters are those allowing a reasonable balance between a too narrow focus of the search process and a too weak guidance.

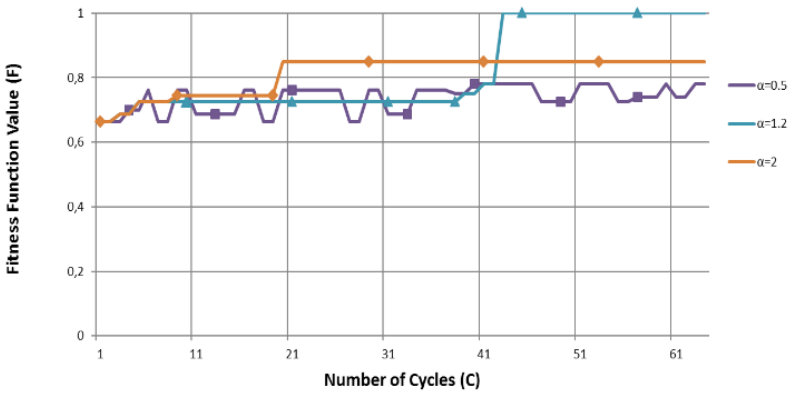

Figure 7. Fitness Function evolution for different values of α

6. CONCLUSION

In this paper the new approach to breaking Simplified-AES was presented using ant colony optimization. This method occurred to be quite effective and promising. The experimental results show that our approach is significantly faster and requires a smaller number of known plaintext-ciphertext pairs, when compared to others attacks.

ACO provides a very powerful tool for the cryptanalysis of Simplified-AES, it is interesting to be applied to cryptanalysis of some others strong encryption algorithms like DES (Data encryption Algorithm) or AES (Advanced Encryption Standard).

REFERENCES

[1] Salabat Khan, Armughan Ali and Mehr Yahya Durrani, "Ant-Crypto, a Cryptographer for Data Encryption Standard," *IJCSI*, vol. 10, no 1, Jan 2013.

[2] Hicham Grari, Ahmed Azouaoui Khalid Zine-Dine, "A Novel Ant Colony Optimization Based Cryptanalysis of Substitution Cipher," *International Afro-European Conference for Industrial Advancement AECIA*, 2016.

[3] M. A. Musa, E.F. Schaefer, S. Wedig, "A Simplified AES algorithm and its linear and Differential Cryptanalysis", *Cryptologia*, pp.148-177, Apr 2003.

[4] H. K. Bizaki, S. David Mansoor, A. Falahati, "Linear Cryptanalysis on Second Round Mini-AES", *International Conference on Information and Communication Techmologies*, 2006, pp. 1958-1962.

[5] S. D. Mansoori, H. Khaleghei Bizaki, "On the Vulnerability of Simplified AES Algorithm Against Linear Cryptanalysis," *International Journal of Computer Science and Network Security*, vol. 7, no. 7, pp. 257- 263, 2007.

[6] Simmons S., "Algebraic cryptanalysis of simplified AES," *Cryptologia*, vol. 33, no. 4, pp. 305–314, 2009.

[7] Vimalathithan M. Omana, C. Metra, "Cryptanalysis of Simplified-AES Encrypted Communication," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 13, no. 10, Oct 2015.

[8] Valarmathi M. L., Vimalathithan, R., "Cryptanalysis of simplified-aes using particle swarm optimization," *Defence Sci. J.*, vol. 62, no. 2, 117–121, 2012.

[9] Rania Saeed (B) and Ashraf Bhery, "Cryptanalysis of Simplified-AES Using Intelligent Agent," *10th International Conference, HAIS 2015*, Bilbao, Spain, June 22-24, 2015.

[10] M. Dorigo, V. Maniezzo, A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 26, no. 1, pp. 29-41, 1996.

[11] M. Dorigo, L. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp.1, pp. 53 -66, 1997.

[12] T. Stutzle and H. Hoos, "Improvements on the ant system, introducing the MAX-MIN ant system," in *Proc. ICANNGA97—Third Int. Conf.Artificial Neural Networks and Genetic Algorithms*, Wien, Germany: Springer-Verlag, 1997.

BIOGRAPHIES OF AUTHORS

Hicham Grari is a PhD student within the LAROSERI Lab. at Faculty of Sciences – Chouaib Doukkali University, El Jadida/Morocco. He holds an Engineer Degree in Computer Science in 2005. His doctoral research investigates the use of metaheuristics in cryptology field.

Ahmed Azououi received his license in Computer Science and Engineering in June-2001 and Master in Computer Science and Telecommunication from University of Mohammed V - Agdal, Rabat, Morocco in 2003. He received his PHD in Computer Science in 2014 and Engineering at Department of Computer Science, ENSIAS (National School of Computer Science and Systems Analysis), Rabat, Morocco. Currently, he is an Associate Professor at Department of Computer Science, Faculty of sciences, University Chouaib Doukkaly, El Jadida, Morocco. His areas of interest are Information systems, Coding Theory and Artificial Intelligence.

Khalid Zine-Dine received his PhD degree from the Mohammed V University of Rabat, Morocco, in 2000. He spent four years in bank Information System as a network & system security project manager. Currently, he is an Associate Professor and lead of LAROSERI Lab. at Faculty of Sciences – Chouaib Doukkali University, El Jadida/Morocco. His research interests are in the area of wireless ad hoc and sensor networks, Mobility, cloud computing, microgrid and system & network architectures and protocols. Dr. Zine- Dine was a co-organizer and co-chair of conferences and is involved in more than 06 PhD Thesis and funded research projects.