Ant-Crypto, a Cryptographer for Data Encryption Standard

Salabat Khan, Armughan Ali and Mehr Yahya Durrani

**Dept. of Computer Science, COMSATS Institute of Information Technology,
Attock Campus, Pakistan**

Abstract

Swarm Intelligence and Evolutionary Techniques are attracting the cryptanalysts in the field of cryptography. This paper presents a novel swarm based attack called Ant-Crypto (Ant-Cryptographer) for the cryptanalysis of Data Encryption Standard (DES). Ant-Crypto is based on Binary Ant Colony Optimization (BACO) i.e. a binary search space based directed graph is modeled for efficiently searching the optimum result (an original encryption key, in our case). The reason that why evolutionary techniques are becoming attractive is because of the inapplicability of traditional techniques and brute force attacks against feistel ciphers due to their inherent structure based on high nonlinearity and low autocorrelation. Ant-Crypto uses a known-plaintext attack to recover the secret key of DES which is required to break/ decipher the secret messages. Ant-Crypto iteratively searches for the secret key while generating several candidate optimum keys that are guessed across different runs on the basis of routes completed by ants. These optimum keys are then used to find each individual bit of the 56 bit secret key used during encryption by DES. Ant-Crypto is compared with some other state of the art evolutionary based attacks i.e. Genetic Algorithm and Comprehensive Binary Particle Swarm Optimization. The experimental results show that Ant-Crypto is an effective evolutionary attack against DES and can deduce large number of valuable bits as compared to other evolutionary algorithms; both in terms of time and space complexity.

Keywords: *Ant-Crypto, Binary Ant Colony Optimization, Comparison of Optimization Techniques, Cryptanalysis of Data Encryption Standard.*

1. Introduction

Most important and precious element in any Information/ Communication system is DATA. Apart from giving us information and knowledge about past events/ activities and patterns, analysis of data can also help us in decision making process, keeping in view the objectives to be achieved in future. There are numerous techniques to store, retrieve and mine the data in databases and data warehouses but in this competitive world where adversaries can illegally access the data, the only way to survive and compete the adversaries is to keep the valuable data, safe and secure. The data cannot be kept secure using classical security techniques e.g. locks; either physically or electronically. In the literature, two inevitable categories of attacks are described; one is passive attack and the other is active attack. In the passive attack, an attacker get access to the communication system and find information contained within secret data. These attacks are difficult to intercept because the attacker do not change the contents of the original data. On the other hand, in active attack an attacker not only gets access to the data but also disrupt the original data. The active attacks are easily detectable but difficult to recover.

Organizations cannot rely on the original form of their secret data and they even don’t want any attacker to launch the passive attack (active attack is more harmful) against their communication/ information system. So, they use encryption schemes usually known as cipher (encryption algorithm) in the field of cryptography. Some ciphers e.g. Data Encryption Standard (DES), Advance Encryption Standard (AES) uses secret keys to encrypt the secret data/ message or plaintext. Ant-Crypto is a novel swarm based attack for the cryptanalysis of DES. Cryptanalysis is about the techniques in cryptography that tries to recover the original message or plaintext from an encrypted message, without knowing the secret key used during encryption phase. It includes the study of mathematical techniques e.g. linear cryptanalysis and differential cryptanalysis for attacks against communication/ information system security.

There are two types of ciphers based on the unit of a plaintext that goes under processing; first, the Block ciphers and second, the Stream ciphers. Block ciphers are modern ciphers and operates on a block or chunk of the original plaintext using fixed transformation based on the combination of substitution and permutation. Stream ciphers process a single byte of a message at a time when en/decrypting. DES is based on feistel block cipher. Substitution ciphers are easily breakable due to their weedy encryption process [13]. The length of the key is the main indicator of how difficult it would be to break a cipher. DES with a 56 bit key length makes brute force attack infeasible as it would take several years to find the secret key even if the original plaintext is known. In the next section, we will review the related work in the domain of DES cryptanalysis.

2. Related Work

Cryptanalysis of DES is an interesting problem for the researchers in the field of cryptography. The effectiveness of optimization techniques in cryptanalysis is apparent with the research carried out on classical as well as modern block ciphers (that are more resistant to attacks). Spillman et al. [1] and Castro et al. [2] used Genetic Algorithm in their research to break different ciphers. A fair amount of detailed analysis of how different optimization techniques can be used in the field of cryptography is provided in the research of Clark [3]. Further, Clark et al. [4] have investigated the automated cryptanalysis of classical ciphers which is also considers as an extensive effort. In their thesis [5], they also investigated the effectiveness of simulated annealing, tabu search and genetic algorithm for the cryptanalysis of substitution ciphers. Garici et al. [6] used a population based approach for the automated cryptanalysis of substitution ciphers.

All the papers described above are considered effective [7] for classical ciphers because the complexity of these ciphers is low and there is some inherent linear relationship that may be exploited by an attacker to break them easily. Modern ciphers are complex and highly resistant to any known attack for classical ciphers e.g. character frequency analysis and information of digram and trigram etc. DES also experiences the avalanche effect; a property of any encryption algorithm such that a small change in either the plaintext or the key produces a significant change in the ciphertext. The strength of DES is discussed in detail [8] against differential cryptanalysis attacks. Matsui [9] presented the first experimental cryptanalysis of DES using an improved version of linear cryptanalysis technique. Bafghi et al. [10] proposed weighted directed graph model to find the differential characteristics of a block cipher using Ant Colony Optimization based on shortest path. Laskari et al. [11] used Particle Swarm Optimization technique for the cryptanalysis of simplified version (four-rounded) of DES. J. Song et al. [12], [7] used Genetic Algorithm for the cryptanalysis of two and four-rounded DES. Waseem Shahzad et al. [13] used comprehensive learning binary particle swarm optimization for the cryptanalysis of four rounded DES. In this paper, we use for the first time, Ant Colony Optimization algorithm for the cryptanalysis of four-rounded DES (block cipher). The remainder of this paper is organized as follows.

In the next section, we review the working of DES. In Section 4, we present the basics of ant colony optimization meta-heuristic. In Section 5, architecture, design and detail of the proposed solution is given. Subsequently, in Section 6, we present some experimental results of Ant-Crypto compared with other techniques to show the promising ability of our approach. Finally, Section 7 will conclude this work.

3. Four Rounded DES

In feistel ciphers, transformations are usually carried out as a combination of substitution and permutation. A mapping function is applied repeatedly several times in an iterative manner; iteration is usually called a round. DES is one of the most famous feistel ciphers and has enjoyed widespread use internationally during last few decades. It consists of sixteen rounds and operates over a 64 bits data block using a 56 bits key. DES is a symmetric cipher as encryption and decryption is almost same but the same secret key is applied in reverse order during decryption. Four-Rounded DES is a restricted form of the original DES in which only four rounds are used during encryption/ decryption.

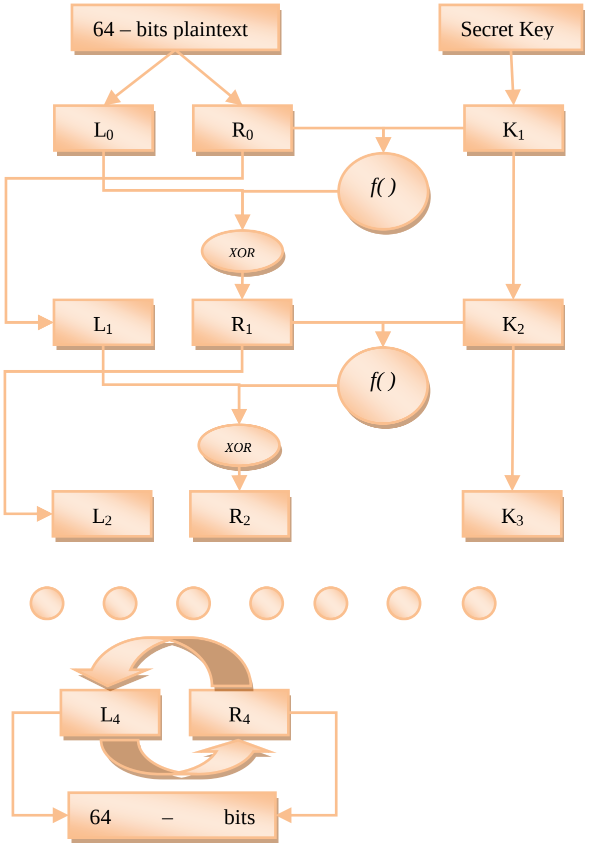

Fig. 1 Four Round DES Working

In DES, 64 bits data is divided into left and right halves. The key is stored as 64 bits but reduced to 56 bits after applying a permutation table. This 56 bits key is then divided in two parts each of 28 bits. After that 16 sub keys are created after applying circular left shifts and permutations according to the given shift and permutation tables, respectively. In each round of DES, a main function is applied to the right half of data and a subkey of 48 bits. During this process, eight S-boxes are used which convert each 6-bit block into a 4-bit block generating 32-bit data. Finally, the left half of the data is XORed with 32-bit output of the main function. In each round, two mapping equation are used; first $L_i = R_{i-1}$ and second $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$ for above-mentioned process. Where 'i' denotes the round number and $\oplus$ denotes the XOR operation. Li and Ri are left and right halves, respectively. Ki is the ith subkey used in round 'i'. In four-rounded DES, maximum value of 'i' will be four. The readers are directed to [14], [15] for more detailed description of Data Encryption Standard (DES).

4. Ant Colony Optimization

Suppose, we have a connected graph G = (V, E) where |V| denotes the total number of nodes/ vertices and |E| total number of connecting edges in graph. The simple ant colony optimization meta-heuristic can be used to find the shortest path between a given source node 'Vs' and a given destination node 'Vd' in the graph 'G'. Each edge of the graph connecting the nodes 'Vi' and 'Vj' has a variable (artificial pheromone), which is modified by the ants when they visit the nodes [19].

From a node, when an ant decides which node to move next, it uses two parameters to calculate the probability of moving to a particular node; first, distance to that node and second, amount of pheromone on the connecting edge. Let $d_{i,j}$ be the distance between the nodes 'i' and 'j', the probability that the ant chooses 'j' as the next node after it has arrived at node 'i' where 'j' is in the set 'S' of nodes that have not been visited [19] is:

$$p_{i,j} = \frac{[\tau_{i,j}]^{\alpha}.[\eta_{i,j}]^{\beta}}{\sum_{k\epsilon S}[\tau_{i,k}]^{\alpha}.[\eta_{i,k}]^{\beta}} \qquad (1)$$

Where $\tau_{i,j}$ is the pheromone/ trail value on edge and $\eta_{i,j}$ is a heuristic value calculated as $1/d_{i,j}$. The parameters α and β are influencing factors of pheromone value and heuristic value, respectively. The pheromone on edges is modified using equation (2) as:

$$\tau_{i,j} = \tau_{i,j} + (Q/L) \qquad (2)$$

Where 'Q' is some constant and 'L' is the length of the tour, small the value of 'L' high the pheromone value added to the previous pheromone value on an edge. With time, concentration of pheromone decreases due to diffusion affects; a natural phenomenon known as evaporation. This also ensures that old pheromone should not have a too strong influence on the future. Evaporation can be performed using equation (3).

$$\tau_{i,j} = \tau_{i,j}.\rho \quad (where\ \rho\ is\ between\ 0\ and\ 1) \qquad (3)$$

5. Proposed Technique

In the following subsections, each and every stage of Ant-Crypto is further discussed in a fair amount of detail:

5.1 Search Space (A directed graph)

The cores of our approach include "structure of search space" and "calculation of heuristic value". The search space modeled in the article is generic in nature; as it can very easily be used for the cryptanalysis of other ciphers including but not limited to DES and AES. The search space consists of two layers of vertices. One layer at top and second at bottom, both consists of '*n*' vertices where '*n*' is the length of secret key. In our case, the key length is 64 as used by DES. For the cryptanalysis of AES, the '*n*' will be 128. The top layer vertices are labeled as '1' where the bottom layer vertices are labeled as '0'; thus called Binary Ant Colony Optimization.

In order to precisely describe the constraints on the movement of ant, let us see the search space from another point of view. Search space is a grid of two rows and '*n*' columns. Every vertex in a column is connected to all the vertices in the next column through directed edges except the vertices in last column, so, the total number of edges are *4*(n-1)* and total number of vertices are *(n*2)*. An ant starts it tour from a node at left most column by choosing the node label '0' or '1', randomly. An ant can only move from left to right and its tour is finished at the *nth (i.e. the last)* column. In a column, an ant can only select a single vertex during a particular tour. At the end, when the tour is completed, it will consist of '*n*' vertices labels which in turn form an *n*-bit long binary string. This binary string is a candidate or guessed key that will be applied to the original plaintext and a candidate cipher text is calculated.

5.2 Initialization

At start of ACO, edges in the search space are required to be initialized with some small values of pheromone. Usually this is done randomly but in our case, initialization of pheromone is not random. A seeding population is generated based on the equation (4). Seeding population is then used to initialize the pheromone values.

$$M_k \oplus C_k \qquad (for\ k = 1,2,3.....n) \tag{4}$$

Where '*k*' is the *kth* bit of plaintext *M* and its corresponding ciphertext is denoted as *C*. Seeding population is generated based on random multi plaintext-ciphertext pairs. XOR operation is basically performed on a plaintext and its correspondence ciphertext. This will speed up the evolution process [7] and is adopted from the approximate expression in linear cryptanalysis [20, 21]. Note that equation (4) will result in *n*-bit long binary string denoting '*n*' vertices and the edges between these vertices will be initialized with some small pheromone values. In order to keep diversity in our search space, we used seeding population of size 100. We used four ants as original swarm size and the parameters 'α' (pheromone influence factor) is set to '1.5' and 'β' (heuristic influence factor) is set to '1' in our experiments.

5.3 Fitness Function

Let '*n*' is the key length, C_{si} and C_{ti} are the *ith* bits of the original ciphertext generated using original secret key and the candidate ciphertext generated using trial key; then fitness function '*f* ' is defined as:

$$f(Cs,Ct) = \frac{\sum_{i=1}^{n} h(Cs_i,\ Ct_i)}{n}$$

$$h(Cs_i, Ct_i) = \begin{cases} 1, & \text{if } Cs_i = Ct_i \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The possible range of fitness value is between 0 – 1. The tour of an ant nearest to the real key is supposed to have higher fitness value, see e.g. [13].

5.4 Heuristic Value

The transition probability equation (1) needs a heuristic value calculation method from the problem domain as an efficient search methodology. Ant-Crypto uses no heuristic value in the first iteration (discussed in section 5.5). After *1st* iteration, four ants results in four candidate keys. The candidate key with the best fitness value using equation (5) is saved as a global best ant. Now, in the subsequent iterations, at every decision point, ant uses heuristic value which is calculated as follows:

$$\eta_{i,j} = f(Original\_Key, Concatenate(\lambda_{|1\ to\ i|}, j, \Omega_{|i+2\ to\ n|}))$$

The 'Concatenate' is a function that returns an *n*-bit binary string after concatenating the given three binary string parameters. The '$\lambda_{|1\ to\ i|}$' is a binary string denoting the partial tour of an ant where '*i*' is the vertex in the *ith* column (in the search space) at which ant has to decide which node to move next. The '*j*' is the vertex in next column where an ant can move, only two values are possible i.e. either '0' or '1'. The '$\Omega_{|i+2\ to\ n|}$' is the best ant binary substring from index '*i+2*' to '*n*'. So, concatenated binary string becomes a guessed key which is evaluated using equation (5) to be used as a heuristic value in equation (1). Note that '*n*' is the key length as discussed, previously.

5.5 Proposed Algorithm

Seeding population is generated and initialization is done as discussed in Section 5.2. The ants complete their tours by making decisions using equation (1). Each completed tour represents a trial/ candidate key to the problem. Pheromone values are updated using equation (6), only best ant in a particular iteration is allowed to update the pheromone values on the edges constituting the tour. The ants also update the best ant information based on their tours fitness values. Evaporation is performed after an iteration using equation (3). The pheromones over the edges constituting the tour of an ant is updated using equation (6), so larger the fitness value, the greater is the amount pheromone concentrated, and the more attractive the edges become for subsequent ants:

$$\tau_{i,j} = \tau_{i,j} + \frac{tour\ fitness}{log_2 n} \tag{6}$$

We used 500 runs (R) and in each run there are 1000 iterations (N). In a run during an iteration, if we found the fitness value of the best ant greater than or equal to a threshold value 'γ', we declare the tour (64 bits binary string) an optimum key. Once the optimum key is found next run is started. Several optimum keys are generated across multiple runs. For all optimum keys, we count the sum of 1 and 0 for all bit positions and each sum is then divided by *R* (denoting total candidate optimum keys) [13]. Now, if the sum after division is higher than 'Ϛ' (a threshold already set by user) then these bits can be deduced as '0' or '1'. We fix these bits in the seeding population and start next run. The algorithm runs again and again until all bits are deduced.

Table 1: Comparison of ACO with PSO and GA for four rounded DES

Rounds	γ	ς	Opt.rate (GA)	Opt.rate (PSO)	Opt.rate (Ant-Crypto)	Suc.bits (GA)	Suc.bits (PSO)	Suc.bits (Ant-Crypto)
Four	0.78	0.60	0.87	0.93	**0.94**	4	4	**05**
Four	0.80	0.70	0.38	0.47	**0.50**	13	19	**19**
Four	0.80	0.75	0.38	0.47	**0.49**	8	16	**17**
Four	0.82	0.65	0.12	0.32	**0.35**	9	13	**15**

Suppose that we have three candidate keys; Key 1: (11010101), Key 2: (11001000) and Key 3: (10001110), ‘R’ = 3 and the value of ‘ς’ is 0.80. So, the count (1) is (3,2,0,1,2,2,1,1) and count (0) is (0,1,3,2,1,1,2,2). Now, Count (1)/ R = (**1**, 0.67, 0, 0.33, 0.67, 0.67, 0.33, 0.33) and Count (0)/ R = (0, 0.33, **1**, 0.67, 0.33, 0.33, 0.67, 0.67). For all the bit positions in the first vector Count(1)/ R and in second vector Count(0)/ R if values are greater than or equal to ‘ς’, they are fixed as 1 and 0, respectively. In the above example, bit number one of the seeding population is fixed as 1 and bit number three is fixed as 0. In the successive runs, these deduce bits will initialize the edges with high pheromone values and help in guessing other valuable bits of the complete secret key. Note that however, deduction may be either true or false; if true, subsequent runs will produce good results and more valuable bits of the secret key will be found. Otherwise, ants will be attracted to local optimum peaks because of this deceptive deduction. The pseudo code of the algorithm is given below. This algorithm is run again and again until all the bits of the secret key are found.

Proposed Algorithm

1. Generate seeding population from multi plaintext-ciphertext pairs and perform initialization
2. Complete the tours of ants by making the decisions using probability equation (1)
3. Calculate fitness value for the tours of ants according to equation (5)
4. Update best ant information. Update pheromone using equation (6)
5. Perform evaporation using equation (3)
6. Repeat this process until an optimum key is found or maximum number of iterations (*N*) have been reached, if found optimum key, deduce bits and fixed these bits in seeding population
7. Repeat the steps from 1 – 6 for (*R*) times run

6. Simulation Results

The performance of the proposed algorithm is compared with the cryptanalysis of four-rounded DES using Genetic Algorithm [7] and Binary Particle Swarm Optimization [13]. We have implemented the Ant-Crypto in Visual Studio 2008 C#. We used the optimum rate (the ratio of the optimum keys in all solution) and number of success bits (the bits in guessed key that are matched with the original key) as the performance metrics.

Table 1. shows the parameter values used during experiments. It also summarizes the performance of our approach against other evolutionary attacks against DES. The fixed number of generations ‘N’ was 1000 and run ‘R’ was 500 in our experiments. Variations of the values of parameters ‘γ’ and ‘ς’ are also shown in Table 1. The reason to use these parameter values is to compare the Ant-Crypto with GA [7] and Binary PSO [13] where the same parameters values were used. We can see that the Ant-Crypto performs better than the results obtained using GA and Binary PSO. We obtained higher number of optimum keys and success bits on different setting of the parameters values. We found largest number of optimum keys when we settled small value of ‘γ’; as the target for ants was easy to find. We can see from Table 1. that when ‘γ’ is 0.80 and ‘ς’ is 0.70, we have maximum number of success bits i.e. 19 in total. The algorithm when run four times with these parameter values, finds all bits of the secret key.

Table 2: Experimental results for 1, 2 and 3 Rounded DES

Rounds	γ	ς	N	R	Opt.rate (Ant-Crypto)	Suc.rate (%) (Ant-Crypto)
One	0.70	0.25	10,000	100	**0.97**	**0.99**
Two	0.75	0.30	10,000	150	**0.96**	**1.00**
Three	0.76	0.30	10,000	200	**0.96**	**0.99**

Table 2 summarizes the results obtained for one, two and three rounded DES. The ‘N’ i.e. total numbers of generations are 10,000 in order to get the optimum key with high probability during a run as we are using small numbers of runs ‘R’, this time. The values of ‘γ’ and ‘ς’ in experiments are used such that the results can be found quickly with good possible quality. From Table 2, it can be observed that the approach is performing very well in case of optimum success rate, as DES with one, two and three rounds is easy to break. Each round in DES adds more complex mapping between plaintext and ciphertext. So, it will be a kind of a complex challenge to use this approach further for the cryptanalysis of more rounds of DES as compared to less complex four-rounded DES.

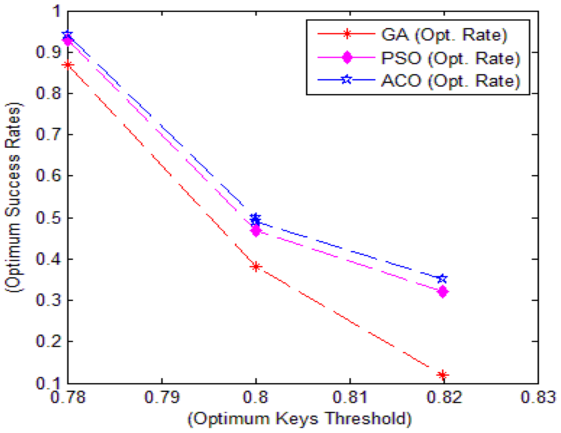

Fig. 2 Comparison of GA, B-PSO and B-ACO based on optimum success rate

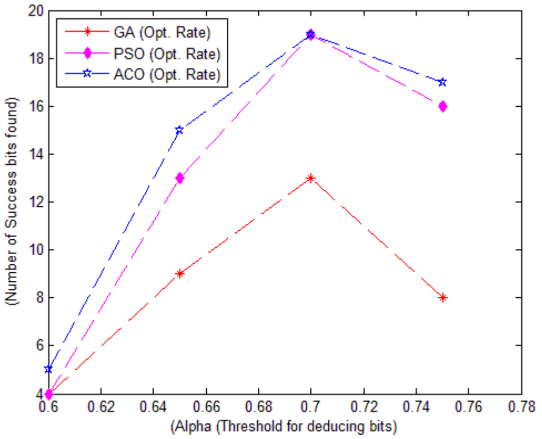

Fig. 3 Comparison of GA, B-PSO and B-ACO based on success bits

In Fig. 2, the comparison of GA, PSO and Ant-Crypto (ACO) is given for optimum success rate. X-axis shows the optimum key threshold and Y-axis shows the optimum success rate. Graph depicts that the Ant-Crypto performs better than GA as well as PSO based cryptanalysis of four-rounded DES in all the cases. Fig. 3 shows the comparison of GA, PSO and Ant-Crypto for finding the success bits. X-axis shows the threshold of guessing a bit and Y-axis shows the number of success bits. Comparison results show that Ant-Crypto is effective and robust technique for cryptanalysis of modern block cipher such as DES as compared with other optimization techniques e.g. GA and PSO. Fig. 2 depicts that Ant-Crypto obtained highest optimum success rate with optimum key threshold is 0.78 and Fig. 3 depicts that Ant-Crypto has highest number of success bits with Alpha (threshold for deducing bits) is equal to 0.70.

7. Conclusions

This article proposed a new version of cryptanalysis algorithm for four rounded DES using binary ant colony optimization. Ant-Crypto is compared with other two well known evolutionary algorithms (GA and PSO) used for cryptanalysis of four rounded DES. We compared the results on that basis of optimum rate and number of success bits found. The experimental results show that Ant-Crypto is an efficient and effective method for the cryptanalysis and it achieves higher optimum rate and number of success bits when compared with other evolutionary approaches used for the cryptanalysis of four rounded DES.

There are several important avenues for future research; the search space structure and/ or heuristic value calculation may be changed for acquiring other valuable findings. It will be a good idea to use a hybrid solution e.g. combining Genetic Algorithm and Ant Colony Optimization (if possible) etc. and have some new experiments. There are several parameters that are required to be well tuned in order to get these results and thus results may be improved further for example, the values of parameters alpha and beta can be tuned in an effort to find better values than those currently used in our experiments (i.e. $\alpha = 1.5$, $\beta = 1$). Furthermore, the effect of pheromone evaporation rate ‘ρ’ needs to be studied in search of an optimal value. Currently we are, somewhat arbitrarily, using $\rho = 0.15$. There are different variants of the original ACO algorithm that can be used to obtain better results. This approach can also be applied to cryptanalysis of some other block ciphers e.g. AES.

References

[1] R. Spillman, M. Janssen, B. Nelson, and M. Kepner. Use of A Genetic Algorithm in the Cryptanalysis of simple substitution Ciphers, April 1993, Vol. 17(1), pp. 31-44.
[2] Julio César Hernández Castro, José María Sierra, Pedro Isasi and Arturo Ribagorda. Genetic Cryptoanalysis of Two Rounds TEA. ICCS2002. Lecture Notes In Computer Science; Vol. 2331 2002. pp. 1024-1031.
[3] Clark. Modern Optimization Algorithms for Cryptanalysis. Proceedings of Second IEEE Australian and New Zealand Conference on Intelligent Information Systems. 1994.
[4] Clark and Ed Dawson. Optimization Heuristics for the Automated Cryptanalysis of Classical Ciphers. In Journal of Combinatorial Mathematics and Combinatorial Computing, Papers in honour of Anne Penfold Street, 1998, vol. 28, pp. 63-86.
[5] Andrew John Clark. Optimization Heuristics for Cryptology, PhD thesis, 1998.
[6] Mohamed Amine Garici, Habiba Drias. Cryptanalysis of Substitution Ciphers Using Scatter Search. IWINAC 2005. Lecture Notes in Computer Science Vol. 3562. pp. 31-40.

[7] J. Song, H. Zhang, Q. Meng and Z. Wang. Cryptanalysis of Four-Round DES Based on Genetic Algorithm. International Conference on Wireless Communications, Networking and Mobile Computing WiCom 2007. Issue. 21-25. Sept. 2007, pp. 2326 – 2329.

[8] Don Coppersmith. The Data Encryption Standard (DES) and its strength against attacks. IBM Journal of Research and Development. Vol. 38 Issue 3. May 1994. pp. 243– 250.

[9] M. Matsui. First Experimental Cryptanalysis of the Data Encryption Standard. Advances in Cryptology—CRYPTO '94, Lecture Notes in Computer Science, Vol. 839 pp 1-11.

[10] Abbas Ghaemi Bafghi, Babak Sadeghiyan. Finding Suitable Differential Characteristics for Block Ciphers with Ant Colony Technique. Proceedings of Ninth International Symposium on Computers and Communications 2004 (ISCC"04), Vol. 2 pp .418-423.

[11] E.C. Laskari, G.C. Meletiouc, Y.C. Stamatiou, M.N. Vrahatis. Evolutionary Computation based Cryptanalysis: A first study. Nonlinear Analysis, 2005, pp. 823-830.

[12] Jun Song, Huanguo Zhang, Qingshu Meng, Zhangyi Wang. Cryptanalysis of Two-Round DES using Genetic Algorithm. ISICA 2007, LNCS 4683, pp. 583–590, 2007.

[13] Waseem Shahzad, Abdul Basit Siddiqui, Farrukh Aslam Khan. Cryptanalysis of Four-Rounded DES using Binary Particle Swarm Optimization. GECCO'09, Montréal Québec, Canada. ACM, pp. 2161-2166, 2009.

[14] National Bureau of Standards, Data Encryption Standard, FIPS-Pub.46. National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.

[15] Whitfield Diffie, Martin Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard", IEEE Computer 10(6), pp74-84, June 1977.

[16] M. Dorigo. Optimization, "Learning and Natural Algorithms". PhD thesis, 1992.

[17] L.M. Gambardella and M. Dorigo. Ant-Q: A Reinforcement Learning Approach to the TSP. In Proceedings of Twelfth International Conference on Machine Learning, pp. 252-260, 1995.

[18] L.M. Gambardella and M. Dorigo. Solving Symmetric and Asymmetric TSPs by Ant Colonies. IEEE International Conference on Evolutionary Computation, pp. 622627, 1996.

[19] S. Khan, Mohsin Bilal, M. Sharif, Malik Sajid, Rauf Baig. Solution of n-Queen Problem Using ACO. International Multitopic Conference, Islamabad, IEEE, pp. 1-5, 2009.

[20] Mitsuru Matsui: Linear Cryptanalysis Method for DES Cipher, pp. 386-397, 1993.

[21] Mitsuru Matsui: The First Experimental Cryptanalysis of the Data Encryption Standard. CRYPTO, pp. 1-11, 1994.

[22] S. Khan et. al., Cryptanalysis of Four-Rounded Data Encryption Standard using Binary Ant Colony Optimization", ICISA, IEEE, pp. 1-7, 2010.