International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

BAT algorithm for Cryptanalysis of Feistel cryptosystems

T. Mekhaznia[1]*, A. Zidani[2]

Accepted 15th August 2014 **DOI: 10.18201/ijisae.82426**

Abstract: Recent cryptosystems constitute an effective task for cryptanalysis algorithms due to their internal structure based on nonlinearity. This problem can be formulated as NP-Hard. It has long been subject to various attacks; available results, emerged many years ago remain insufficient when handling large instances due to resources requirement which increase with the amount of processed data. On another side, optimization techniques inspired by swarm intelligence represents a set of approaches used to solve complex problems. This is mainly due to their fast convergence with a consumption of reduced resources. The purpose of this paper is to provide, and for a first time, a more detailed study about the performance of BAT algorithm in cryptanalysis of some variant of Data encryption standard algorithms. Experiments were performed to study the effectiveness of the used algorithm in solving the considered problem and underline the difficulties encountered.

Keywords: *Cryptanalysis, Feistel ciphers, bat algorithm.*

1. Introduction

The *cryptanalysis* is a manner of studying cryptosystems, ciphers and related concepts in order to break them without prior knowledge of encryption algorithms or the right way of decryption. This fact presents a hard task against research in data security. Its principle lies in use of mathematical tools necessary to provide the right attack [1]. The brute force is the most popular attack; it attempts to guess all possibilities which conduct to solution; the technique is sure but requires much processing time and so, less success in practice. Other alternatives were available in literature such *linear* and *differential* cryptanalysis which are able to break a wide variety of ciphers, nevertheless, and given their reduced setting, they remain ineffective against modern cryptosystems.

Feistel ciphers are algorithms for symmetric key encryption scheme which operate on large blocks of data by using fixed transformations, they are known for their high level of security against various attacks. This is due to their internal structure which based on high nonlinearity and low autocorrelation. Among them, the *Data Encryption Standard*, an algorithm characterized by its simplicity of implementation, high speed of encryption [2] and resistance against various attacks [3].

Research in this area, are actually intended to heuristic algorithms which appear the favour tool to break complex ciphers. Many works [4] [5] shown that algorithms based *swarm intelligence* have a successful potential to handle wide instances and may be adapted to produce an approximate solutions for a large variety of optimization problems.

Swarm intelligence algorithms are well-known meta-heuristics that were successfully used to approximate solutions of various real-world optimization problem with minimal resources consumption [6]. They offer an intelligent system which distributed functioning an independence of movement of agents which will replaces preprogramming and centralized control. In last years, many of such algorithms were emerged [7].

In this paper, an approach of *Bat algorithm* (BAT) is used for breaking some variants of Feistel ciphers. The main reason for using BAT, is its global solution finding property, it uses few parameters without any need to give initial approximation to unknown parameters. It has been successfully applied in a wide range of research application areas. It is proved that it gets better results in a faster and cheaper way compared with other methods.

2. Swarm intelligence

An idea introduced by [8], *Swarm intelligence* heuristics (SI, in short) are evolutionary nature inspired metaheuristics that satisfy at least one of the two goals of information technology research, namely the generation of solutions with maximum benefit by using a minimum of resources, however, no proof of the optimality of the solution can be confirmed, given that research becomes useless, if in a exploration space, a cross between the local and global solution occurs [9]. SI heuristics uses a basic population of individuals which represents candidate solutions. They evolve by a decentralized and self-organized system according to a common rule. This principle allows treating very large space of solutions but do not guarantee if an optimal solution is ever found.

Nature inspired heuristics, part of evolutionary computation heuristics, are stochastic algorithms which took their inspiration from social comportment of animals living in large communities such bird flocking, fish schooling or ant colonies. They are based on a population of individuals that interact and evolve according to common rules. This principle is used to produce algorithms which resolve complex tasks without centralized control.

3. BAT Algorithm

Bat Algorithm [10], is a new metaheuristic population based approach, inspired from the hunting behaviour of bats. In their

[1] *LAMIS Laboratory, University of Tebessa, Algeria.*
[2] *Department of Computer Sciences, University of Batna, Algeria*
** Corresponding Author:Email: mekhaznia@yahoo.fr*
Note: *This paper has been presented at the International Conference on Advanced Technology&Sciences (ICAT'14) held in Antalya (Turkey), August 12-15, 2014.*

flying and, in order to avoid obstacles and detect prey or their roosts in dark, bats emit a bisonar [11] throughout their environment, the echo bounces permit to identify kinds of surrounding objects. Studies [12] show the loudness of emitted pulse varies from lowness rate when searching to loudest when homing toward prey or their roosts with a decreasing duration of sounds.

Similar to other nature inspired algorithms such ACO, AG and PSO, BAT has been implemented for continuous optimization problems where possible solutions are represented by bat's positions.

The principle of BAT is illustrated by following steps:

- In a search space $\mathbb{R}^n$, and at time t, each bat i has a position x_i^t and a velocity $\boldsymbol{v}_i^t \in \mathbb{R}^n$.
- In their randomly fly with a constant velocity v, each bat i emit a uniformly pulse frequency f_{min}.
- At the perception of a pry, parameters are adjusted depending on the distance to pry according to following relations:

$$f_i = f_{min} + \beta(f_{max} - f_{min}) \quad (1)$$

$$v_i^{t+1} = v_i^t + f_i(x_i^t - x_g^t) \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$

where β a random vector distributed in range [0,1], xg the position of best bat of swarm. The algorithm presents BAT is as follows:

Algorithm 1. BAT

Input: BatNumber(N_g >2), ObjectiveFunction
Output: OptimalSolution
Initialization :
  Generate Bat parameters $\boldsymbol{x_i}$, $\boldsymbol{v_i}$, $\boldsymbol{c_i}$ (i=1.. N_g)
  Evaluate Bat solutions S($\boldsymbol{x_i}$), (i=1.. N_g)
  Initialize g ←{x_k / S(x_k) ←min(S(x_i), i=1.. Ng}
  Initialize exit criterion T←0, Iter←0,T_{ref}, S_{ref} and $Iter_{max}$
While not ($\boldsymbol{Iter_{max}} < Iter$ and $\boldsymbol{T_{ref}} < T_{process}$ and $S_{ref} > S(g)$)
  Pick random numbers: β ∈ [0,1]
  For each *bat* *i* do
    $f_i = f_{min} + \beta(f_{max} - f_{min})$ with $\boldsymbol{f}_i \in \{\boldsymbol{f_{min}}, \boldsymbol{f_{max}}\}$
    $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \boldsymbol{f_i}(\boldsymbol{x}_i - g)$ with $\boldsymbol{v}_i \in \{\boldsymbol{v_{min}}, \boldsymbol{v_{max}}\}$
    $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i$
    If $(S(\boldsymbol{x}_i) < S(\boldsymbol{g}))$
      $g \leftarrow \boldsymbol{x}_i$
      increase rate pulse and reduce loudness
    Endif
  Endfor
  Update *Iter* and $T_{process}$
EndWhile
Report g and S(g).

4. Feistel ciphers

Based on *confusion* and *diffusion* principle, the Feistel ciphers (FC in short) is a special class of iterated block ciphers which maps a plaintext to a ciphertext by a sequential r times repetition of a nonlinear transformation (called round function).

In order to produce a ciphertext C from a plaintext M, Feistel cryptosystem proceeds for each bloc of n-bits of M, r iterations of the round function fi using several keys. Initially, each block is split into two halves L0 and R0 of n/2 bits each. At iteration i, the round function is applied to one half using a subkey, the output is exclusive-ored with the other half. The two halves are then swapped as shows in following relation:

$$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus f_i(R_{i-1}, k_i)) \quad (4)$$

where f_i (i>0), a nonlinear function usually represented as a substitution boxe (called *sboxe)*. It substitutes an input of n bits size with an output of m bits size (m<n).

The advantage of the algorithm is that the encryption and decryption functions are identical. To reverse a round, it is only necessary to apply the same transformation again, which will cancel the changes of the binary operation XOR. The FC becomes the basis for many encryption schemes, among them the DES is the most popular one.

4.1. Data Encryption Standard

The most widely used Feistel block cipher cryptosystem is the Data Encryption Standard (DES or 16DES in short) which transform 64-bit input block in a series of steps into similar size output block through a 16-round process. The round function fi uses 8 nonlinear sboxes. Each of them is a substitution which mapping 6 to 4 bits. Detailed description of DES is given in [13] [14] [15].

The strength of DES lie in the nonlinearity induced by the sboxes, it remains an important model for the construction of secure block ciphers. Also, the brute force attack need an average of 2^55 tries which takes high time complexity such that, the resources for search in an acceptable period are not available. Its major weakness is out of the 2^56 keys there are four weak keys, six semi weak keys and 48 possible weak keys. The weak key is that, after parity drop operation, consist of all 0, all 1 or half 0 and half 1 [9].

4.2. Four Rounded DES

The Four rounds DES algorithm (4DES in short) [16] has the same properties as DES algorithm. It uses an fk function which merge substitution, combination and logic addition that take place in four rounded with four subkeys by using the same table of expansion and permutation (sboxes) as DES.

The 4DES algorithm is less complex than DES but remain difficult to break.

4.3. Simplified DES

The Simplified DES (SDES in short) is a variant of DES introduced by [17]. It has similar properties and structure as DES with much smaller parameters.

The SDES encryption algorithm uses an 8 bit block of plaintext and a 10-bit key as input, and produces an 8-bit block of ciphertext as output. The decryption algorithm takes an 8 bit key of ciphertext and the same 10 bit key as input and produces an 8 bit block of plaintext.

The encryption algorithm is performed in five steps: (1) an initial permutation, (2) a complex function fk, where both permutation and substitution operations are performed based on the key input, (3) a permutation of the two halves of the data, (4) again performing the function fk, and (5) inverse initial permutation. The same steps are followed for the decryption operation in a reverse order. The algorithm uses two reduced sboxes (size 4x4) and an expansion/permutation table (size 1x8).

5. Proposed approach

The approach proposed is based on the framework of the original bat algorithm by considering the general characteristics of whole species of bats, we redefine the corresponding operation to the bats' behaviors as presented in the following pseudo-code:

Algorithm 2. Proposed approach

Input: ProblemDimension (>2), ObjectiveFunction
Output: OptimalSolution
Initialization :
Generate x_i, v_i, f_i (i=1.. N_g)
Evaluate S(x_i), (i=1.. N_g)
Initialize p ← x_i et g ←{ x_k / S(x_k) ←min(S(p)} (i=1.. N_g, k=1.. N_p)
Initialize T←0, Iter←0,T_{ref}, S_{ref} and $Iter_{max}$
While not ($Iter_{max} < Iter$ and $T_{ref} < T_{process}$) and $S_{ref} > S(g)$)
Pick random numbers: β ∈ [0,1]
$f_i = f_{min} + \beta(f_{max} - f_{min})$ with $f_i \in \{f_{min}, f_{max}\}$
$v_i \leftarrow v_i + f_i (x_i - p_i)$ with $v_i \in \{v_{min}, v_{max}\}$
$x_i \leftarrow x_i + v_i$
If $(S(x_i) < S(x_p))$ $p_i \leftarrow x_i$ *Endif*
If $(S(p_i) < S(g))$ $g \leftarrow p_i$ *Endif*
Endfor
Update *Iter* and $T_{process}$
EndWhile
Report g and S(g).

6. Problem formulation

The character frequency analysis is the process of determining at which frequency a symbol (in general an alphabet letter) of the encrypted message occurs within the ciphertext. This fact can be used along with the knowledge of character frequency within the language used. In English, for example, the single character (1-gram) the most frequent is 'E' with an occurrence of 12% in a text, followed by 'T' with 8%, while the 'Z' occurs with only 0.05%. Similarly, the group of two characters (2-gram) most common are 'TH, HE, AN, ..'. The triplet (3-gram) the most common is 'THE'. These statistics have been developed based on Corpus [18] [19] and illustrated on tables called 'letter frequency table' [20] [21].

The natural way to prove the effectiveness of a candidate key used in decryption is to compare the frequency character analysis of the decrypted text to the frequency analysis of the language used. The fitness function is built about this idea. It can have different forms which are used on several combination schemes of n-gram analysis. Various alternatives forms of this function were available in literature [22] [23] [24] [25]. The most commonly used is given by following equation:

$$F(k) = \alpha \sum_{i=1}^{26} \left| D(i) - C(i) \right|^{u} + \beta \sum_{i,j=1}^{26} \left| D(i,j) - C(i,j) \right|^{b} . \quad (5)$$

where D, C denotes respectively known language statistics and decrypted text statistics. u, b: denotes respectively 1-gram and 2-gram statistics, α and β (with α+β=1) are weights assigning different priorities to 1-gram and 2-gram and k, the key used in decryption process

7. Experiment environment and results

The keys used for decryption are initially generated randomly. Each key represents a possible solution. At each iteration (assumed as a generation), the population of particles or individuals is evolved by application of transformations to keys according to following rule: a move of bat from position i to a new position j corresponds to a flip of j-th bit of key to 1 (if vi > 0.5) and to 0 otherwise. In order to avoid the bat swarm explosion, the velocity values may be in range of]0,1]. The obtained keys are then valued according to fitness function used. Only the best keys may survive. The process will continue until it reaches an acceptable decrypted text, unless, it stopped after a fixed number of iterations.

The experiments have been conducted in order to outline the performance of the proposed algorithm on a set of sample various texts extracts from ICE [20] and ciphered by SDES, 4DES and DES cryptosystems using several keys. The decryption algorithm is coded on Matlab 2.14. Initially, treatment starts with feasible solutions generated randomly. A total number of 10 runs is performed for each ciphertext.

The results obtained after carrying the mentioned experiments are illustrated below. The objective of tests is to analyze the effectiveness of the used algorithm of each variant of DES algorithms. Figure 1 shows the percentage of recovered bits of encryption key for each variant when using a fixed processing time of 150 seconds.

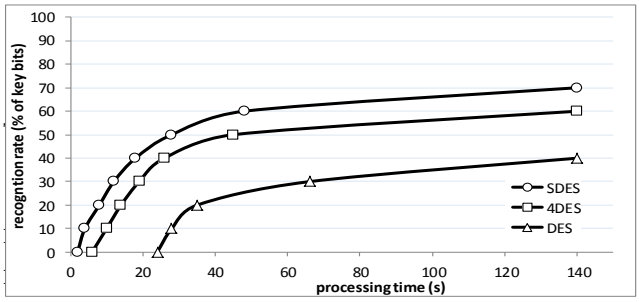
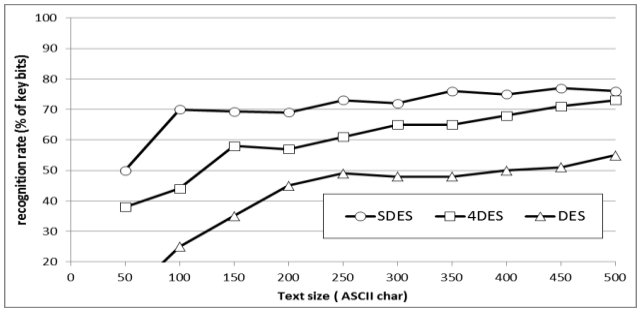


Figure 1. Percent. of recovered correct bits via processing time.

The second test summarizes the improvement of the algorithm



when using various sizes of ciphertexts:

Figure 2. The improvement of the algorithm

8. Conclusion and future work

In this paper, we have introduced an approach of BAT algorithm for cryptanalysis of some variants of Feistel ciphers as an optimization problem. The experiments show that the algorithm can be successfully applied in resolve of such problem. The overall average results show that the algorithm allow locating the full key (in case of reduced DES) and more than 50% of bits-key (in case of DES) with acceptable resource consumption. The tests were operates on a reduced space of data, however, we think that is possible to improve the presented results by a well choice of problem parameters such environment constants, ciphered data and languages statistics. In addition, the study may open avenues to investigate the effectiveness of other evolutionary computation techniques for further attacks of more complicated cryptosystems.

References

[1] S. Rao & al. (2009). Cryptanalysis of a Feistal Type Block Cipher by Feed Forward Neural Network Using Right Sigmoidal Signals. International Journal of Software Computing, Vol.4(3).

[2] S.Ali K, Al-Omari Putra Sumari. (2010). Spiking Neurons

with ASNN BASED-Methods for the Neural Block Cipher. International journal of computer science & information Technology. Vol.2(4).

[3] R. Singh, D. B. Ojha. (2010). An Ordeal Random Data Encryption Scheme (ORDES). International Journal of Engineering Science and Technology. Vol. 2(11). Pages.6349- 6360.

[4] C. Blum, X. Li, (2007). Swarm intelligence in optimization', natural Computing Series, Springer.

[5] T.S.C. Felix, M.K. Tiwari. (2007). Swarm Intelligence, Focus on Ant Particle Swarm Optimization. Int. Tech Education and Publishing..978-902613-09-7.Austria.

[6] A. Gherboudj, S. Chikhi. (2011). A modified HPSO Algorithms for Knapsack Problem. CCIS. Springer.

[7] G.S. Sharvani, N.K. Cauvery, T.M. Rangaswamy. (2009). Different Types of Swarm Intelligence Algorithm for Routing. International Conference on Advances in Recent Technologies in Communication and Computing.

[8] Beni, G., Wang, J. (1989). Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy.

[9] J. Olamaei, T. Niknam, G. Gharehpetian. (2008). Application of particle swarm optimization for distribution feeder reconfiguration considering distributed generators. AMC. Pages 575-586.

[10] X. S. Yang. (2010). A New Metaheuristic Bat-Inspired Algorithm. Nature Inspired Cooperative Strategies for Optimization.

[11] D. R. Griffin. (1958). Listening in the dark. Yale Univ. Press. New York.

[12] J. R,. Speakman, P. A. Racey. (1991). The cost of being a bat. Nature. V 350. Pages. 421−423

[13] S. Ghorui & al. (2000). A simpler and elegant algorithm for computing fractal dimension in higher dimensional state space. PRAMANA Journal of Physics. Indian Academy of Sciences. Vol 54(2), L331–L336.

[14] W. Stallings. (2008). Cryptography and Network Security Principles and Practices. Pearson Education.

[15] A. B. Forouzan. (2008). Cryptography and Network Security. Tata McGraw hill Education, 2nd ed.

[16] E.C. Laskari & al. (2005). Evolutionary computation based cryptanalysis: A first study. Nonlinear Analysis: Theory, Methods and Applications. Vol. 63. Pages. e823–e830.

[17] E. Schaefer. (1996). A Simplified Data Encryption Standard Algorithm. Cryptologia. Vol. 20(1). Pages. 77-84.

[18] Robert, L. (2000). Cryptological Mathematics. The Mathematical Association of America. NY.

[19] Nelson, G., Wallis, G. and Bas, A. (2000). Exploring Natural Languag: Working with the British Component of the International Corpus of English. John Benjamins Publishing Company. Amsterdam.

[20] Singh, S. (1999). The code book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography. Doubleday. New York, NY, USA, 1st edition.

[21] Beker, H. and Piper, F. (1982). Cipher Systems: The Protection of Communications. John Wiley & Sons.

[22] Jakobsen, T. and Knudsen, L.R. (2001). Attacks on block ciphers of low algebraic degree. J. of Cryptology. Vol. 14(3). Pages. 197-210.

[23] Nalini, N. and Raghavendra, G. (2006). Cryptanalysis of Simplified Data Encryption Standard via Optimisation Heuristics. Int. J. of Computer Science and Network Security. Vol. 6(1B). Pages.240-246.

[24] Verma, A. K., Dave, M. and Joshi. R. C. (2007). Genetic Algorithm and Tabu Search Attack on the MonoAlphabetic Subsitution Cipher in Adhoc Networks. Journal of Computer Science. Vol. 3 (3). Pages. 134-137.

[25] Ganapathi, S. and Purusothaman, T. (2011). Reduction of Key Search Space of Vigenere Cipher Using Particle Swarm Optimization. J. of Computer Science. Vol 7(11). Pages. 1633-1638.