International Journal of Computer Theory and Engineering, Vol. 2, No. 3, June, 2010
1793-8201

Nomadic Genetic Algorithm for Cryptanalysis of DES 16

S.Siva Sathya, T.Chithralekha and P.AnandaKumar

***Abstract*-Key determination in the Cryptanalysis of DES-16 is considered to be a NP hard problem that involves a large search space. Multi-population Genetic Algorithms are considered to be more effective than single population genetic algorithm for such types of problems in obtaining the optimal solution in lesser time. In this direction, this paper presents a novel multi-population genetic Algorithm called Nomadic Genetic Algorithm (NGA) for breaking the encrypted message created using full 16 round Data Encryption Standard (DES) in less than 2 36 complexities. A comparison of the performance of Standard GA (SGA) and NGA is also presented. The performance of the algorithm is found to be better and considerably faster than exhaustive search and other existing GA.**

***Index Terms*-Genetic Algorithm, Nomadic Genetic Algorithm, Cryptanalysis, DES, Multi-population GA.**

I. Introduction

Cryptanalysis is the art of deciphering encrypted communications without knowing the actual way to decipher the crypt message. It is one of the major challenging areas of intense research in the discipline of security. It is a process of looking for weakness in the design of ciphers. A cryptosystem takes as input a plain text and a known key and produces an encrypted version of the plain text known as the cipher text. Data Encryption Standard (DES) is a block cipher [8], [12] with a 64-bit block size which uses 56-bit keys. This makes it susceptible to exhaustive key search with modern computers and special-purpose hardware. DES is still strong enough to keep most random hackers and individuals out, but it is easily breakable with special hardware by government, criminal organizations, or major corporations. Many attempts had been performed for breaking the DES [9].

In the previous versions of the attack on DES, researchers have made use of brute force or exhaustive-key search. The attacks were prone to higher time complexity, space complexity, and less success rate. To overcome the above drawbacks Genetic Algorithms were made use of. GA had been explained by Holland [11] as an adaptive heuristic search method that depends on the evolutionary ideas of natural selection and genetics. The basic goal of a genetic algorithm is to simulate the process of natural evolution, taking into consideration the principle of survival of the fittest [1]. It is generally used in situations where the search space is relatively large and cannot be traversed efficiently by classical search methods.

Actually, they have been recently successfully applied to the cryptanalysis of simple substitution, transposition, knapsack ciphers, and for Cryptanalysis of DES-8 respectively. As DES systems have a large key space and it is impossible to find out the encryption key using traditional search algorithms, a novel genetic algorithm with its own chromosome representation and fitness evaluation has been attempted and implemented. Though it proved better than existing algorithms, to further improve the performance, a multi-population GA was used and the obtained results were compared with that of SGA.

The organization of the rest of the paper is as follows: Sectuion-2 describes the DES Cryptanalysis in detail. Section-3 explains related work for cryptanalysis of DES. Section-4 details the proposed GA and NGA. Section -5 gives the simulation results and section-6 conclude.

II. DES Cryptanalysis

Differential cryptanalysis is a method which analyses the effect of particular differences in known plain text pairs on the differences of resultant cipher text pairs. These differences can be used to assign probabilities to the possible keys and to locate the most probable key. This method usually works on many plain text pairs with same particular differences of the resultant cipher text pairs. For DES-like cryptosystems the difference is chosen as a fixed XOR ed value of the two plain texts.

Though there are several types of attack, the focus of this paper is on the known plain text attack [12] and Differential cryptanalysis (DC) [8], [9], [14] technique. Known plain text attack means the, cryptanalyst knows some pairs of the plain text and the full cipher text using DC technique,

Differential cryptanalysis can be done in three steps:

- Analyzing the cipher pairs.
- Using Cipher analysis to construct differential characteristics.
- Key Bit extraction.

A. Analyzing the cipher pairs

Let x' and x'' be the pairs in the known plain text.
Let y' and y'' be the pairs of cipher text.
Now the attacker finds out the right pair, by finding differences among the pairs.

$$\Delta x = x' XOR \quad x'' \qquad \Delta y = y' XOR\, y''$$

Now, the right pairs are selected based on the highest probability between the pairs (Δx, Δy), where Δx is the input difference and Δy is the output difference of the corresponding pairs.

B. Using cipher analysis to construct differential characteristics

After making the cipher analysis, the next process is the construction of Difference distribution table. This table displays the distribution of keys, by the definition of DC. The most probable key will occur at the same position, and the wrongs keys will occur randomly. Eliminating the randomly distributed keys, and considering the most probable ones, we could proceed to the next step.

C. Differential characteristics

The idea is to attack the cipher by finding a subset of key bits following the (n-1)th round. Differential characteristics involve plain text bits and input data. By picking up the highest differential pair from the distribution table, the attacker should discard the zero difference pair, considering non-zero differential pair alone.

D .Key Bit Extraction

For n rounds (16 rounds), we can obtain the key in (n-1)th round ie.,15th round, but not completely. Attack involves the pair wise decryption of the last round cipher and testing the corresponding input to find out the right pair (non zero input differential pair to the expected (s boxes). The last round is decrypted as follows:

- XOR with the sub key.
- Check for non zero differentials.
- Repeat step1, until step 2 is satisfied.

At the end of each round, the sub key which has the greatest value is considered to be the correct sub key. Once some keys are found we can make use of any search method like exhaustive key search, brute force attack or Genetic algorithm to find out the optimum key bits.

III. Related Work

Several techniques are found in literature for the cryptanalysis of DES. First cryptanalytic attack was performed in the year 1977. The drawback of this attack is the time complexity. Later, Diffie and Hellman suggested exhaustive search of the entire key space on a parallel machine. The problem with this is the cost spent on the VLSI chip. Hellman suggests a special purpose machine which produces 100 solutions per day with an average wait of one day. He estimates that the machine costs about $4-million and the cost per solution is about $1-100 million. The pre-processing is estimated to take around 2.3 years on the same machine. Chaun and Evertse showed a meet in the middle attack. They proved that a meet in the middle attack of this kind is not applicable to DES reduced to eight or more rounds.

Biham and Shamir introduced a new kind of attack that can be applied to many DES-like iterated cryptosystems.[8], [9] This is a chosen plaintext attack which uses only the resultant cipher texts whose plaintexts can be chosen at random, as long as they satisfy the difference condition, and the cryptanalyst does not have to know their values. The attack is statistical in nature and can fail in rare instances.

Mohammed Hasan, the author has proposed a Genetic Algorithm for cryptanalysis of DES 8", [10]. In this, the researchers have made use of two methods; one is Stored Pair Cryptanalysis (SPCA) and the other is Generated Pair cryptanalysis (GPCA). In SPCA, the right pairs are stored and the fitness values are computed on the stored right pairs. Other genetic operators are applied on the right pairs and finally, the key is computed. The drawback of this approach is the excessive amount of memory required for storing the pairs. GPCA does not use any stored right pairs and hence memory requirement is not high. It uses a hamming distance based fitness evaluation scheme, but the number of key bits obtained is not noteworthy. From the study of the literature it is inferred that GA can be either combined with differential cryptanalysis method or relied upon solely to break down block-ciphered texts.

IV. Proposed System Design

The proposed system design is based on the following idea. The total number of right pairs, np, is computed "differentially" and stored in the memory. The problem of using a huge number of right pairs can be solved by generating the right pairs genetically. Such generation process is carried out by exploiting the relation Y = P xor X. Thus if P is available, then Y can be obtained when X is genetically generated. The pair that satisfies the underlying S-boxes may be an expected key. Thus the proposed system works as follows:

Step 1: Apply the DC techniques as explained in section 2.

Step 2: The most probable keys are obtained from the step 1, and stored for future processing.

Step3: Apply genetic algorithm to find out the most probable keys.

Step 4: Find the key bits and check for termination condition.

A. Proposed Genetic Algorithm

Genetic Algorithms are randomized procedures that work on the principle of genetics and natural evolution. Here GA has been used to find the most probable key. Thus the input to the GA is: Stored probable key bits obtained from step-2 in section 2.The expected output of GA is : Some known bits of the 48 bit key. (This being a known plain text attack). The GA procedure is given below:

Step 1: The most probable keys obtained from step-2 forms the seed generating the initial population. Consider each S-Box as a single chromosome.

Step 2: Evaluate fitness for each chromosome.

- According to the definitions of DC, the most probable key will occur at the same place repeatedly. Based on the above, the fitness value is calculated.
- For each chromosome, repeated values that occur in S-Boxes (S1– S8), is considered to be the most probable one.
- Store the fittest value for each chromosome.

Step 4: Apply selection procedure.

Step 5: Apply cross over operation.

Step 6: Place the key bits found by comparing it with the (n-1) th i.e., 15th round key using the DC technique.

Step 7: Repeat step 3, until all the 48 key bits are

obtained, or the fitness value doesn't improve over 'n' user defined consecutive generations, where 'n' may be 10 or 20.The Chromosome representation, fitness evaluation and the choice of the Genetic Operators play a vital role in this proposed GA and are described below:

B. Chromosome Representation

Each chromosome is represented as a one dimension array. The array takes integer value. For example, there are 8 S-Boxes in DES Algorithm. Each is considered as a single chromosome.

03
0F
1E
1F
2A
2B
37
3B

Fig 1: Chromosome Representation

C. Initial Population

Each chromosome is also called as individual. The initial population will consist of the required number of individuals. This is based on the problem size & may be specified by the user. For example fig. 2 shows a sample.

S1	S2	S3	S4	S5	S6	S7	S8
60	61	32	55	51	63	49	50
37	62	40	61	55	63	59	53
47	55	61	57	30	63	56	63
56	51	60	48	15	61	63	10
49	60	49	33	12	30	62	59
59	63	63	68	14	34	59	11
45	53	12	15	23	35	61	52
58	23	14	19	23	31	49	56

Fig 2: Initial Population

D. Fitness Evaluation

Fitness Procedure

According to the definitions of DC, the most probable key will occur at the same place repeatedly. Based on the above, the fitness value is calculated.

- For each chromosome, repeated values that occur in S-Boxes (S1– S8), are considered to be the fittest.
- Store the fittest value for each chromosome.

For example consider a single chromosome S1 from the above fig 2.

60	61	60
37	**37**	**37**
47	63	47
56	56	42
49	50	49
59	58	59
45	43	45
58	60	58
1(a)	**1(b)**	**1(c)**

Fig 3: Fitness evaluation Procedure

Fig. 1 (a) is the initial position of the key bits for the S-Box 1 and fig. 1(b) shows the position of the keys in the S-box1 after the first generation. Fig.1 (c) shows the positions of the key after a particular number of generations; in this the key 37 is found to be repeated. From the above figures a, b, c, it can be considered that the most number of repeated values will be the most probable one. Hence, it can be inferred that the value 37 is a probable key because it is repeated in the same position after a number of generations.

E. GA Parameters

The following are the GA parameters used during the experimentation:

No. of Individuals	100
Selection	Roulette Wheel
Crossover	Single Point
Crossover%	95%
Elitism%	5%
Termination	Fixed Number (200) generations.

V. Proposed Nomadic Genetic Algorithm

To further improve the performance of the above described GA, a variant of GA, namely NGA has been used to implement the DES-16. This is a multi population GA and has proved to yield much better results for other problems [15], [16]. The algorithm is simple and hence the DES implementation using GA can be easily converted to NGA.

The design involves breaking the population into sub

populations and allowing for migration between the sub groups. The rest of the procedure is same as GA. The representation, fitness evaluation, operators are also same except for the process of evolution which is explained below.

1. Generate initial population randomly
2. Evaluate the fitness of each Individual
3. Sort the Individuals in Non-increasing order of their Fitness Values
4. The population is then arranged into groups based on their fitness range.
   a. Select Individuals from each group
   b. Apply Crossover/Mutation operators
   c. Evaluate the fitness of offspring
   d. Add offspring to the same group.
5. Combine all the groups in to a single list
6. Sort the list in non-increasing order of their fitness values and trim the list to the size of the groups.
7. Repeat the process from Step-4 to the required No. of Generations
8. Select the best (high fit) individual.

NGA gives equal chance of survival to the high fit as well as the low fit individuals. This increases the diversity in the Population and thus avoids local optima, which in turn will lead to a better performance of the GA.

VI. Experimental Results And Analysis

A number of experiments were carried out by giving different inputs and applying GA and NGA for breaking DES. The results are tabulated in table 2.The table below illustrates the key bits found using SGA and NGA for the given input text and known plain text. The table-3 show the performance of NGA in terms of speed and performance , and is given below.

TABLE 2: Key Bits Found Using SGA And NGA

Exp. No	Plain Text	Known Plain Text	Key Bits found using SGA	Key Bits found using NGA
1	A friend's frown is better than a fool's smile.	Frown fool smile	A fr_ _ nd _ f_ _w _ better th_ n a fool's smi_ le	A friend's frown better t_an a fool_ smiles
2	A friend is easier lost than found.	Friend easier lost	A fr _ _ nd is ea_s_er l_st than found.	A frien_ easie_ lost tha_ found.
3	A loveless life is a living death.	Loveless living death	_ Lo_ _ l_ s _ life is a l_v_ n g death.	A Loveles_ life is a livin_ death.
4	Before you meet the handsome prince you have to kiss a lot of toads.	Handsome prince kiss toads	H_ n_ s _ _ me prince K_ _ s toads.	Handsome you meet prince K_ _ s t_ad_.
5	A man of straw needs a woman of gold.	Man straw needs gold	Man Straw n_ _d s gold.	A m_n of straw n__ds a woman of gold.
6	A handsome shoe often pinches the foot.	Handsome shoe pinch	Hand_ s_ me	A ha_ _ _ome shoe often pinc_ _s the foot.
7	As you go through life, make this your goal, watch the dough nut and not the hole.	Through goal watch dough	Thr _ _gh goal watch dog	A_ you go through life, m_ _e this your goal, watch the do_ _h nut and not the h_le.
8	No man is worse for knowing the worst of himself.	Worse knowing worst	Worse Kno_w_n_ worst	No m_n is worse for k___ing the worst of h_mself.
9	After dinner rest a while, after supper walk a mile.	Dinner supper while mile	Dinn_ _ su_ _ er while mile	After di__er rest a while, after supper w_ _k a mile.
10	Good wine ruins the purse, and bad wine ruins the stomach.	Wine ruins stomach	Wi_ _ ruins stomach	G__d wine ruins the p__se, and bad wine ruins the st_ _ach.

TABLE 3: Comparison Of SGA And NGA In Terms Of Performance And Time Taken

Exp. No	No. of Keys found using SGA	No. of Keys found using NGA	Time Taken using SGA	Time Taken using NGA
1	36	42	26 seconds	21 seconds
2	30	42	26 seconds	22 seconds
3	30	36	26 seconds	20 seconds
4	30	36	26 seconds	19 seconds
5	36	30	26 seconds	25 seconds
6	36	42	26 seconds	25 seconds

7	30	36	26 seconds	35 seconds
8	36	36	26 seconds	31 seconds
9	30	42	26 seconds	25 seconds
10	36	36	26 seconds	25 seconds

From table 2 , it is found that both SGA & NGA are able to find most of the given encrypted text. From table 3, it is inferred that NGA works better than SGA in terms of time taken as well as obtaining maximum number of key bits for most of the inputs.

VII. Conclusion

The determination of key bits in the cryptanalysis of DES 16 using a suitable technique is a challenging task. As it involves a large key space GA was chosen to implement the problem in this paper. To further enhance the performance a Multi-population GA namely Nomadic GA was used for the Cryptanalysis of DES 16 problem. The performance of Nomadic Genetic Algorithm over Standard Genetic algorithm was compared in various aspects such as the number of keys obtained and the time required to break DES. From the experimental results and analysis carried out, it is concluded that the proposed GA was able to obtain good number of key bits when compared to the existing work in literature. Taking a step further, NGA was able to obtain a still better performance due to its migration characteristic. In addition to the increase in the number of key bits found, the memory space and time complexity obtained are also noteworthy. Finally, to conclude, the combination of Differential cryptanalysis and NGA, was able to break the DES algorithm to a greater extent in much lesser time.

References

[1] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine learning", Pearson Education, Ninth edition, 2005.
[2] Darrel Whitley, "A Genetic Algorithm Tutorial", Computer Science Department, Colorado State University, Fort Collins, CO 80523.
[3] Ulrich Bodenhofer, "Genetic Algorithms: Theory and Application", Lecture Notes Third Edition-Winter 2003/2004.
[4] Mitchell Melanie, "An introduction to Genetic Algorithms", A Bradford Book, The MIT press, Fifth printing, 1999.
[5] S.Siva Sathya, S.Kuppuswami, K.Rajasekhar, Nomadic Genetic Algorithm for Course Timetabling, report-2006, Proceeding of the international conference on Information, Science, Technology and management (CISTM) 2007,Hydrabad.
[6] E. Biham and A. Shamir,"Differential cryptanalysis of Data Encryption Standard,", pp 487-502,1991.
[7] C,etin Kaya ko,c, ''Differential Cryptanalysis",Oregon State University.
[8] Hasan Mohammed Husein ,Bayoumi I. Bayoumi , Fathy Saad Holail , Bahaa Eldin M. Hasan, and Mohammed Z. Abd El-Mageed ," A Genetic Algorithm for cryptanalysis of DES-8," International Journal of Network Security, pp213-219,2006.[9]Holland, J.H, (1992). Adaptation in natural and artificial systems. Cambridge, MA: MIT Press..
[9] National Bureau of Standards, Data Encryption Standard, Federal Information Processing Standard(FIPS), Publication 46 , National Bureau of Standards, U.S. Department of Commerce , Washington D.C.,1997.
[10] William Stallings, "Cryptography and Network Security," Tata Mc-Graw Hill Publications, 2007.
[11] Shamir,"Differential Cryptanalysis of full 16-round DES", Technion, 1991.
[12] S.Siva Sathya, S.Kuppuswami and K.Rajashekar (2007). Nomadic Genetic Algorithm for Course Time Tabling problem. Proceedings of the International Conference on Science Technlogy and Management (CISTM 07), Hyderabad, India.
[13] Siva Sathya, S., Kuppuswami,S., Syam Babu,K. (2009). Nomadic Genetic Algorithm for Multiple Sequence Alignment. In International Journal of Adaptive and Innovative Systems (IJAIS), Vol.1, No.1, pp 44-59

S. Siva Sathya received her B.Tech and M. Tech in Computer Science and Engineering from Pondicherry University. She also received her Doctorate in Computer Science and Engineering from the same University. She holds the position of Senior Lecturer in the Department of Computer Science, Pondicherry University, Puducherry, India. She has published a number of papers in international conferences and journals and her research interests include genetic algorithms, bioinformatics, grid computing and information security.

T.Chithralekha received her B.Tech and M. Tech in Computer Science and Engineering from Pondicherry University. She also received her Doctorate in Computer Science and Engineering from the same University. She is currently working as a Reader in the Department of Banking Technology, Pondicherry University, Puducherry, India. She has published a number of papers in international conferences and journals and her research interests include information security for Banking & financial sectors, Multi-agent systems and Multilingual systems.

P.Anandakumar recived his B.sc and M.sc in Computer Science from Pondicherry University.