

# Cryptanalysis of S-DES using Genetic Algorithm

Vimalathithan.R<sup>1</sup>, Dr.M.L.Valarmathi<sup>2</sup>

<sup>1</sup>Sri Krishna College of Engineering and Technology/ECE Department, Coimbatore, India

Email: athivimal@gmail.com

<sup>2</sup>Govt. College of Technology/CSE Department, Coimbatore, India

Email: drmlv@gct.ac.in

**Abstract**—Cryptanalysis with genetic algorithm has attracted much interest in recent years. This paper presents an approach for the cryptanalysis of Simple Data Encryption Standard (S-DES) using genetic algorithm. In this paper, cipher text only attack is adopted and variety of optimum keys is produced based on fitness function. S-DES keys can be found faster using the proposed approach. The experimental results indicate that, this is a promising method and can be adopted to handle other complex block ciphers like DES.

**Index Terms**— Cryptanalysis, S-DES, Genetic Algorithm, Plain text, Cipher text, Cipher text attack.

## I. INTRODUCTION

Cryptography is the study of methods of sending messages in disguised form so that only intended recipients can remove the disguise and read the message. The process of converting a plaintext to cipher text is called enciphering and the reverse process is called deciphering. Cryptanalysis is the study of methods for obtaining the meaning of encrypted information, without access to the secret information which is normally required to do so. Cryptanalysis is one of the challenging research areas in the discipline of security. Typically, this involves finding the key that is used for disguising the message. An attack on Cipher text may be of various types. One type of attack namely *Cipher text only attack* is considered in this paper. In cipher text only attack, the encryption algorithm used and the cipher text to be decoded are known to cryptanalyst. This is the most difficult attack among the classes of attacks encountered in cryptanalysis. This paper considers cryptanalysis of S-DES ciphers. Though S-DES is a much simplified version of DES, Cryptanalysis of S-DES will give a better insight into the attack of DES and other block ciphers.

In the brute force attack, the attacker tries every possible key on the piece of cipher text until an intelligible translation of the cipher text into plaintext is obtained. Cryptographic algorithms are almost designed to make the brute force attack infeasible. Generally, the key space considered by any secret key based algorithm is large enough so that it is not possible for an attacker to try every possible key. But, Genetic Algorithm efficiently solves this problem without searching the entire key space.

The main objective in the area of key breaking is to find *sufficiently good* solution that efficiently solves the cryptanalysis problem for the large scale applications.

Hence, the focus of this work is breaking the S-DES key by conducting a directed random search of a key space using Genetic Algorithm. Since Genetic algorithms are considered as one of the most efficient search techniques, this paper considers Genetic algorithm as a tool to solve the key breaking problem.

The rest of the paper is organized as follows: Section II discusses the earlier studies and works done in this area. Section III presents a brief overview of S-DES and Section IV gives the overview of Genetic Algorithm. Experimental results are discussed in Section V. Finally, Conclusion and Future work are presented in section VI.

## II. RELATED WORK

Several solutions have been proposed in this area. In 1993, for the first time, the paper by Spillman [1] presented a genetic algorithm based approach for the cryptanalysis of substitution cipher. The paper has explored the possibility of random type search to discover the key (or key space) for a simple substitution cipher.

In the same year Mathew used an order based genetic algorithm for cryptanalysis of a transposition cipher. In 1993, Spillman [2] successfully applied a genetic algorithm approach for the cryptanalysts of a knapsack cipher also. In 2006, Garg studied that the efficiency of genetic algorithm attack on knapsack cipher can be improved with variation of initial entry parameters.

In 2006, Garg [3] study gives the base that genetic algorithm can be used to break S-DES. In 2008 Garg [4] explored the use of memetic algorithm to break a simplified data encryption standard algorithm.

In 2006, Nalini [5] compared the attack of SDES using Optimization Heuristics technique and GA based techniques. The results show that GA based approach minimizes the time complexity.

## III. THE S-DES ALGORITHM

This section briefly gives the overview of S-DES Algorithm. The SDES encryption algorithm takes 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 block of ciphertext and the same 10-bit key used for encryption as input and produces the original 8-bit block of plaintext as output. The encryption algorithm uses five basic functions: 1. An initial permutation (IP). 2. A complex function called  $f_K$  which involves both permutation and substitution operations and depends on a key input 3. A simple

permutation function (SW) that switches the two halves of the data. 4. The function  $f_K$  again and 5. A permutation function that is the inverse of the initial permutation (IP-1). The function  $f_K$  takes as input the data passing through the encryption algorithm and an 8-bit key.

A. Key Generation

For key generation, a 10-bit key is considered from which two 8-bit subkeys are generated. In this case, the key is first subjected to a permutation  $P_{10} = [3\ 5\ 2\ 7\ 4\ 10\ 1\ 9\ 8\ 6]$ , then a shift operation is performed. The numbers in the array represent the value of that bit in the original 10-bit key.

The output of the shift operation then passes through a permutation function that produces an 8-bit output  $P_8 = [6\ 3\ 7\ 4\ 8\ 5\ 10\ 9]$  for the first sub key (K1). The output of the shift operation also feeds into another shift operation and another instance of  $P_8$  to produce the second sub key K2. In all bit strings, the leftmost position corresponds to the first bit. The block schematic of the S-DES Key generation algorithm is shown in Fig. 1.

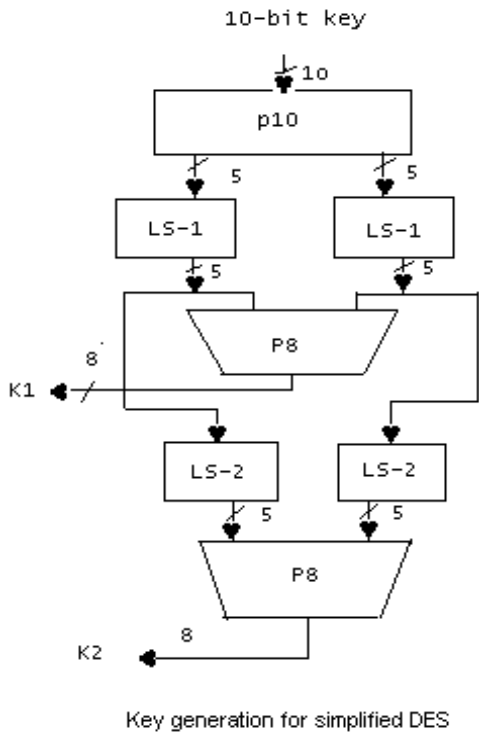


Figure 1. Key Generation

B. Encryption Algorithm

The block schematic of the SDES encryption algorithm is shown in Fig. 2.

The Encryption process involves the sequential application of five functions:

1. Initial and final permutation (IP):

The input to the algorithm is an 8-bit block of plaintext, which is first permuted using the IP function  $IP = [2\ 6\ 3\ 1\ 4\ 8\ 5\ 7]$ . This retains all 8-bits of the plaintext but mixes them up. At the end of the algorithm, the

inverse permutation is applied; the inverse permutation is done by applying,

$$IP^{-1} = [4\ 1\ 3\ 5\ 7\ 2\ 8\ 6]$$

Where,  $IP^{-1}(IP(X)) = X$ .

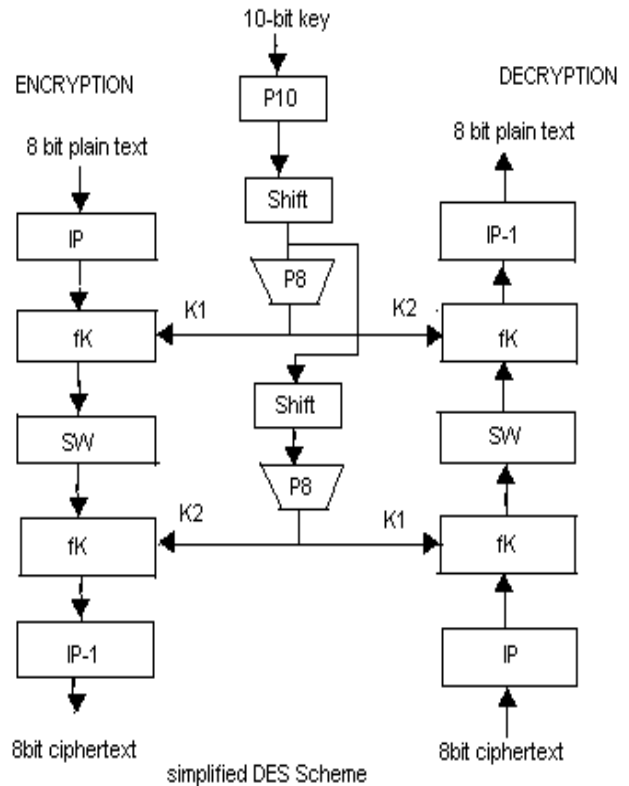


Figure 2. S-DES Encryption

2. Function  $f_K$ :

The function  $f_K$ , which is the complex component of S-DES, consists of a combination of permutation and substitution functions. The functions are given as follows:

$$f_K(L, R) = (L \text{ XOR } f(R, \text{key}), R)$$

where, L, R be the left 4-bits and right 4-bits of the input, XOR is the exclusive-OR operation and key is a sub-key.

Computation of  $f(R, \text{key})$  is done as follows.

- i. Apply expansion/permutation  $E/P = [4\ 1\ 2\ 3\ 2\ 3\ 4\ 1]$  to input 4-bits.
- ii. Add the 8-bit key (XOR).
- iii. Pass the left 4-bits through S-Box  $S_0$  and the right 4-bits through S-Box  $S_1$ .
- iv. Apply permutation  $P_4 = [2\ 4\ 3\ 1]$ .

The two S-boxes are defined as follows:

$S_0$	$S_1$
1 0 3 2	0 1 2 3
3 2 1 0	2 0 1 3
0 2 1 3	3 0 1 0
3 1 3 2	2 1 0 3

The S-boxes operate as follows: The first and fourth input bits are treated as 2-bit numbers that specify a row of the S-box and the second and third input bits specify a column of the S box. The entry in that row and column in base 2 is the 2-bit output.

### 3. The Switch Function (SW):

Since the function  $f_K$  allows only the leftmost 4-bits of the input, the switch function (SW) interchanges the left and right 4-bits so that the second instance of  $f_K$  operates on different 4-bits. In this second instance, the E/P, S0, S1 and P4 functions are the same as above but the key input is K2.

## IV. GENETIC ALGORITHMS

Genetic Algorithms provide robust searches in complex spaces. These algorithms are computationally simple and can be easily improved. They are not limited by the restrictive assumptions about the search space. With an initial random population, they efficiently exploit historical information contained in the gene pool to speculate on new searches with an expected improved performance. The differences between genetic algorithms and a normal search are:

- GA can work with a coding of the parameter set, or with the parameters themselves.
- GA searches from a population of points, not from a single point.
- GA uses payoff (objective function) information, not derivatives or other auxiliary knowledge.
- GA uses probabilistic rules.

A simple genetic algorithm that yields good results in many practical problems is composed of three parameters:

- Reproduction
- Crossover
- Mutation

This section briefly explains the parameters used in a GA for optimization.

### A. Genetic Algorithm Parameters

Selection strategies determine which chromosomes will take part in the evolution process. The different selection strategies that are used in GA are Population Decimation, Proportionate Selection and Tournament Selection. Among these selection strategies, Tournament selection is used in this study. In tournament selection two individuals are randomly selected and then the one with the highest fitness 'wins'. This process continues until the required number of chromosomes has been reached. This method produces slightly better solutions as compared to other Selection strategies.

While the selection strategies are involved with selecting which individuals will take part in the evolution process (be parents), the mating schemes select which two parent chromosomes will mate with one another. The mating schemes that exist include Best Mates-Worst, Adjacent Fitness pairing and Emperor Selective mating. The recommended mating scheme is Best Mates-worst. As the name suggest the chromosomes with the highest fitness mates with the chromosomes with the lowest fitness. This mating scheme is used because of the *avalanche effect* [8] in encryption algorithms.

A crossover occurs when two parent chromosomes mate with one another. When this occurs the two parent chromosomes are both dissected at the same predefined crossover point. The two pieces from the first parent chromosome mate with the two complementary pieces from the second parent chromosome, to form two new chromosomes. Allowable input values for crossover point is between 0 and 1 (i.e.  $0 \leq \text{crossover point} \leq 1$ ) where 0 indicates a crossover point that is determined randomly and 1 indicates that no crossover will occur. For example, a crossover point of 0.5 would indicate that the chromosomes would be cut in half. It is recommended that a random crossover point be used, as this has been shown to produce the best results.

A mutation occurs in a chromosome with a small probability of  $P_{\text{mutation}}$ . When a mutation occurs in a chromosome, a random bit in the binary chromosome is inverted. For example, a '1' will be changed to a '0' and vice versa. Mutations are very important as they allow solutions to be explored which may have not previously been in the optimizer search space. A mutation rate of 0.15 (i.e. 15%) has been found to be the optimum value in most of the cases.

### B. Cost Function

The cost function used for the problem considered is given by equation (1) which represents the ngram statistics of the decrypted message, where the language is assumed to be known.

$$CK = \alpha \sum_{(i \in \tilde{A})} |K(i)^u - D(i)^u| + \beta \sum_{(i, j \in \tilde{A})} |K(i, j)^b - D(i, j)^b| + \gamma \sum_{(i, j, k \in \tilde{A})} |K(i, j, k)^t - D(i, j, k)^t| \quad (1)$$

In equation (1),  $\tilde{A}$  denotes the language alphabet i.e., {A,B...Z, \_}, for English where \_ represents the space symbol, K and D denote the known language statistics and decrypted message statistics respectively, and u, b, and t denote the unigram, digram and trigram statistics respectively;  $\alpha$ ,  $\beta$  and  $\gamma$  are the weights assigning different priorities to each of the three statistics where  $\alpha + \beta + \gamma = 1$ . The values of  $\alpha$ ,  $\beta$  and  $\gamma$  are taken as 0.2, 0.4 and 0.4 respectively [5]. When trigram statistics are used, the complexity of the equation (1) is  $O(P^3)$ , where P is the alphabet size. In view of the computational complexity of trigram, only unigram and digram statistics are used.

## V. EXPERIMENTAL SETUP AND RESULTS

Number of experiments is carried out to outline the effectiveness of GA. The experiment was conducted using Matlab 7 in a PIV System. Among the unigrams, bigrams and trigrams, Unigram is more useful and the benefit of trigram over digram is small. For S-DES, the key size required is 10bits. The plain text and cipher text size are 8 bits. Initially, a population size of 40 is taken with a chromosome size of 10 bits i.e., 40 sets of 10 bit keys are taken randomly and the known cipher text is decrypted using the initial keys. The fitness value is

calculated using equation (1). The parameters for GA are as follows:

Initial Population: 40  
 Chromosome Size: 10 bits  
 No. of Generation: 10  
 Selection: Tournament selection  
 Mating Scheme: Best-Worst Mating  
 Crossover Type: Random  
 Mutation: 0.15

The total number of generations taken is 10. The key was recovered on an average of 5 generations. The size of the search space used by GA is only 250 where as in Brute-force attack, it is 1024. Thus, in case of cryptanalysis using GA there is reduction in search space by a factor of 4 approximately. In the worst case the number of generations is increased to 15. The time taken for Brute-force attack to attack S-DES is approximately 9 microseconds if 1 decryption / microsecond was performed. But using GA the time taken is reduced to 4 microseconds. This is actually, a factor of half the time required to attack by Brute Force attack.

Nalini [5] used GA to attack S-DES; the time taken is around 20 min. But in this approach, the time consumption is reduced and it is less than a minute to attack SDES using GA. This confirms that this method of Cryptanalysis of S-DES using GA can be extended to attack DES which uses 64 bit key size. The time taken for attacking the DES using Brute-Force technique is approximately 10 Hours if 56 bit key is considered and the encryption is done at 1 microsecond per encryption. Using GA the time taken can be reduced by at least by a factor of half, the time taken by Brute Force attack.

#### VI. CONCLUSION AND FUTURE WORK

In this paper, Genetic Algorithm for the cryptanalysis of Simplified Data Encryption Standard is presented. The time complexity of the proposed approach has been reduced drastically when compared to the Brute-Force attack. Though SDES is a simple encryption algorithm, this is a promising method and can be adopted to handle other complex block ciphers like DES and AES. The cost function used here can be applied for other block ciphers also. The future works are extending this approach for attacking DES and AES ciphers.

#### REFERENCES

- [1]. Spillman R, Janssen M, Nelson B and Kepner N, "Use of Genetic Algorithm in Cryptanalysis of Simple Substitution Cipher" *Cryptologia*, Vol.17, No.4, pp. 367-377, 1993.
- [2]. Spillman R, "Cryptanalysis of Knapsack Ciphers using Genetic Algorithms", *Cryptologia*, Vol.17, No.4, pp. 367-377, 1993.
- [3]. Garg Poonam, Genetic algorithm Attack on Simplified Data Encryption Standard Algorithm, *International journal Research in Computing Science*, ISSN1870-4069, 2006.
- [4]. Garg Poonam, Memetic Algorithm Attack on Simplified Data Encryption Standard Algorithm, proceeding of

International Conference on Data Management, February 2008, pg 1097-1108 .

- [5]. Nalini, Cryptanalysis of Simplified data encryption standard via Optimization heuristics, *International Journal of Computer Sciences and network security*, vol 6, No 1B, Jan 2006.
- [6]. Davis, L. "Handbook of Genetic Algorithm", Van Nostrand Reinhold, New York, 1991
- [7]. Neal Koblitz "A course in Number Theory and Cryptography" Springer International Edition, 2008.
- [8]. William Stallings "Cryptography and Network Security Principles and Practicess" Pearson Education, 2004.
- [9]. Mitsuo Gen, R.Cheng "Genetic Algorithms & Engineering Optimization" Wiley Series 2000.
- [10]. Michael D.Vose "The Simple Genetic Algorithm" Prentice hall of India, 2004.



Vimalathithan.R is working as a Senior Lecturer in the Department of Electronics and Communication, Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India.

His Area of interests is Cryptography, Cryptanalysis, Electromagnetic Interference and Compatibility. He received his B.E. degree in Electronics and Communication Engineering from Kongu Engineering College, Perundurai, M.E. Degree from Government College of Technology, Coimbatore and pursuing his Ph.D. in Electronics and Communication Engineering, Anna University, Coimbatore. Also he is a member of Cryptology Research Society of India.



**M. L. Valarmathi** is working as an Assistant Professor in the department of Computer Science and Engineering, Government College of Technology, Coimbatore, Tamilnadu, India.

Her research interests are in the area of Optimization techniques, Image Processing, Algorithm Design, Compilers and Network Security. She received her B.E. degree in Electrical and Electronics Engineering, from Alagappa Chettiyar College of Engineering and Technology, Karaikudi, M.E degree in Computer Science and Engineering, Government College of Technology, Coimbatore and PhD in Computer Science and Engineering, Bharathiar University, Coimbatore. She is a member of ISTE. She has published 26 technical papers in National and International Journals and Conferences. She leads and handles the students at UG, PG and PhD levels.