A COMPARISON OF MEMETIC & TABU SEARCH FOR THE CRYPTANALYSIS OF SIMPLIFIED DATA ENCRYPTION STANDARD ALGORITHM

Poonam Garg

Associate Prof., Department of Information Technology, IMT, Ghaziabad, India-201001

E-mail: pgarg@imt.edu

ABSTRACT

Memetic algorithm(MA) is a population based heuristic search approach for optimization problems similar to genetic algorithm. GAs, however, rely on the concept of biological evolution, but MAs, in contrast, and mimic cultural evolution. The cryptanalysis of simplified data encryption standard can be formulated as NP-Hard combinatorial problem. In this paper, a comparison between memetic algorithm and tabu search were made in order to investigate the performance for the cryptanalysis on simplified data encryption standard problems (SDES). The methods were tested and various experimental results indicates that the proposed memetic algorithm is able to produce high quality solutions quickly and it also demonstrate that memetic algorithm performs better than the genetic algorithm for such type of NP-Hard combinatorial problem.

Keywords: *Simplified data encryption standard, Memetic algorithm, Tabu search, Key search space*

1. INTRODUCTION

Cryptanalysis is the process of recovering the plaintext and/or key from a cipher. The cryptanalysis of simplified data encryption standard can be formulated as NP-Hard combinatorial problem. Solving such problems requires effort (e.g., time and/or memory requirement) which increases with the size of the problem. Techniques for solving combinatorial problems fall into two broad groups – traditional optimization techniques (*exact* algorithms) and non traditional optimization techniques (*approximate* algorithms). A traditional optimization technique guarantees that the optimal solution to the problem will be found. The traditional optimization techniques like branch and bound, simplex method, brute force search algorithm etc methodology is very inefficient for solving combinatorial problem because of their prohibitive complexity (time and memory requirement). Non traditional optimization techniques are employed in an attempt to find an adequate solution to the problem. A non traditional optimization technique - memetic algorithm, tabu search, simulated annealing and tabu search were developed to provide a robust and efficient methodology for cryptanalysis. The aim of these techniques to find sufficient “good” solution efficiently with the characteristics of the problem, instead of the global optimum solution, and thus it also provides attractive alternative for the large scale applications. This paper proposes the cryptanalysis of simplified encryption standard algorithm using memetic algorithm and tabu search. These nontraditional optimization techniques demonstrate good potential when applied in the field of cryptanalysis and few relevant studies have been recently reported.

In 1993 Spillman [16] for the first time presented a genetic algorithm approach for the cryptanalysis of substitution cipher using genetic algorithm. He 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 [12] used an order based genetic algorithm for cryptanalysis of a transposition cipher. In 1993, Spillman [17], also successfully applied a genetic algorithm approach for the cryptanalysts of a knapsack cipher. It is based on the application of a directed random search algorithm called a genetic algorithm. It is shown that such a algorithm could be used to easily compromise even high density knapsack ciphers. In 1997 Kolodziejczyk [11] presented the application of genetic algorithm in cryptanalysis of knapsack cipher .In 1999 Yaseen [19] presented a genetic algorithm for the cryptanalysis of Chor-Rivest knapsack public key cryptosystem. In this paper he developed a genetic algorithm as a method for Cryptanalyzing the Chor-Rivest knapsack PKC. In 2005 Garg [2] has carried out interesting studies on the use of genetic algorithm & tabu search for the cryptanalysis of mono alphabetic substitution cipher. In 2006 Garg [3] applied an attack on transposition cipher using genetic algorithm, tabu Search & simulated annealing. In 2006 Garg [4] studied that the efficiency of genetic algorithm attack on knapsack cipher can be improved with variation of initial entry parameters. In 2006 Garg[5] studied the use of genetic algorithm to break a simplified data encryption standard algorithm (SDES). In 2008 Garg[5] explored the use of memetic algorithm to break a simplified data encryption standard algorithm (SDES).

2. SDES DESCRIPTION

The SDES [18] encryption algorithm takes an 8-bit block of plaintext and a 10-bit key as input and produces an 8-bit block of cipher text as output. The decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used as input to produce the original 8-bit block of plaintext. The encryption algorithm involves five functions; an initial permutation (IP), a complex function called f_K which involves both permutation and substitution operations and depends on a key input; a simple permutation function that switches (SW) the two halves of the data; the function f_K again, and 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. Consider a 10-bit key from which two 8-bit sub keys are generated. In this case, the key is first subjected to a permutation P10= [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 P8=[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 and another instance of P8 to produce subkey K2. In the second all bit strings, the leftmost position corresponds to the first bit. The block schematic of the SDES algorithm is shown in Figure 1.

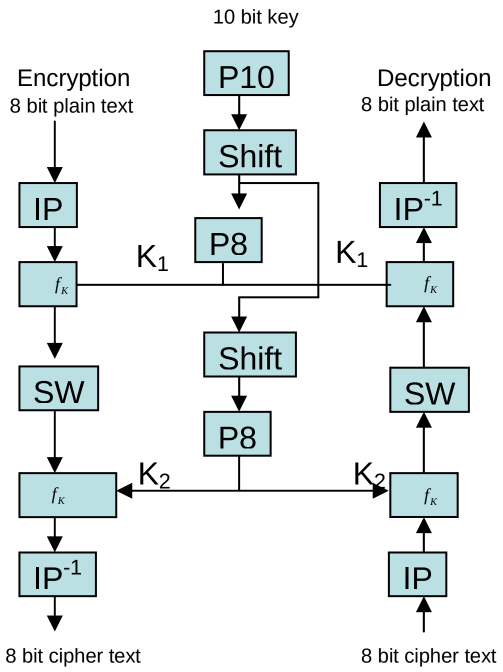

Figure 1: Simplified Data encryption algorithm

Encryption 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 we first permute 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 we have IP^{-1} (IP(X)) =X.
2. The function f_K, which is the complex component of SDES, consists of a combination of permutation and substitution functions. The functions are given as follows.
   Let L, R be the left 4-bits and right 4-bits of the input, then, f_K (L, R) = (L XOR f(R, key), R) where XOR is the exclusive-OR operation and key is a sub - key. Computation of f(R, key) is done as follows.

1. Apply expansion/permutation E/P= [4 1 2 3 2 3 4 1] to input 4-bits.
2. Add the 8-bit key (XOR).

3. Pass the left 4-bits through S-Box S_0 and the right 4-bits through S-Box S_1.
4. Apply permutation P4 = [2 4 3 1].

The two S-boxes are defined as follows:

$$S_0 \quad \begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{pmatrix} \qquad S_1 \quad \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix}$$

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. 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, S_0, S_1 and P4 functions are the same as above but the key input is K2.

3. OBJECTIVE OF THE STUDY

Cryptanalytic attack on SDES belongs to the class of NP-hard problem. Due to the constrained nature of the problem, this paper is looking for a new solution that improves the robustness against cryptanalytic attack with high effectiveness.

The objective of the study is:

- To determine the efficiency and accuracy of memetic algorithm for the cryptanalysis of SDES.
- To compare the relative performance of memetic algorithm with tabu search.

4. METHODOLOGY

4.1 Cost function

The ability of directing the random search process of the tabu search by selecting the fittest chromosomes among the population is the main characteristic of the algorithm. So the fitness function is the main factor of the algorithm. The choice of fitness measure depends entirely on the language characteristics must be known. The technique used by Nalini[13] to compare candidate key is to compare n-gram statistics of the decrypted message with those of the language (which are assumed known). Equation 1 is a general formula used to determine the suitability of a proposed key(k), here ,K is known as language Statistics i.e for English, [A,…….,Z_], D is the decrypted message statistics, and u/b/t are the unigram, bigram and trigram statistics. The values of α, β and γ allow assigning of different weights to each of the three n-gram types where α + β + γ =1.

$$C_k \approx \alpha . \sum_{i \in A} |K^u_{(i)} - D^u_{(i)}| + \beta . \sum_{i,j \in A} |K^b_{(i,j)} - D^b_{(i,j)}| + \gamma . \sum_{i,j,k \in A} \left| K^t_{(i,j,k)} - D^t_{(i,j,k)} \right| \tag{1}$$

When trigram statistics are used, the complexity of equation (1) is O(P3) where P is the alphabet size. So it is an expensive task to calculate the trigram statistics. Hence we will use assessment function based on bigram statistics only. Equation 1 is used as fitness function for tabu search attack. The known language statistics are available in the literature [12].

4.2 Tabu search

The use of the tabu search was pioneered by Glover[9] who from 1985 onwards has published many articles discussing its numerous applications. Others were quick to adopt the technique which has been used for such purposes as sequencing, scheduling, oil exploration and routing.

The properties of the tabu search can be used to enhance other procedure by preventing them becoming stuck in the regions of local minima. The tabu search utilizes memory to prevent the search from returning to a previously explored region of the solution space too quickly. This is achieved by retaining a list of possible solutions that have been previously encountered. These solutions are considered tabu-hence the name of the technique. The size of the tabu list is one of the parameters of the tabu search.

The tabu search also contains mechanism for controlling the search. The tabu list ensures that some solution will be unacceptable; however, the restriction provided by the tabu list may become too limiting in some cases causing the algorithm to become trapped at a locally optimum solution. The tabu search introduces the notion of aspiration criteria in order to overcome this problem. The aspiration criteria over-ride the tabu restrictions making it possible to broaden the search for the global optimum.

An initial solution is generated (usually randomly). The tabu list is initialized with the initial solution. A number of iterations are performed which attempt to update the current solution with a better one, subject to the restriction of the tabu list. A list of candidate solution is proposed in every iteration. The most admissible solution is selected from the candidate list. The current solution is updated with the most admissible one and the new current solutions added to the tabu list. The algorithm stops after a fixed number of iterations or when a better solution has been found for a number of iterations. Figure 2 shows the generic implementation of tabu search.

```
S=GenerateInitialSolution()
InitializeTabuList($TL_1$,……,$TLr$ )
K=0
While termination condition not met do {
AllowedSet(S,k) = { z ∈ N(s) | no tabu condition is violated or at least one
Aspiration criterion is satisfied}
S = BestImprovement(S,AllowedSet(S,K))
UpdateTabuListAndAspirationConditions()
k=k+1 }  end  while
```

Figure 2 : A generic tabu search

4.3 Memetic algorithm

The memetic algorithms [15] can be viewed as a marriage between a population-based global technique and a local search made by each *of* the individuals. They are a special kind of genetic algorithm with a local hill climbing. Like genetic algorithm, memetic Algorithms are a population-based approach. They have shown that they are orders of magnitude faster than traditional *genetic algorithm* for some problem domains. In a memetic algorithm the population is initialized at random or using a heuristic. Then, each individual makes local search to improve its fitness. To form a new population for the next generation, higher quality individuals are selected. The selection phase is identical inform *to* that used in the classical tabu search selection phase. Once two parents have been selected, their chromosomes are combined and the classical operators of crossover are applied to generate new individuals. The latter are enhanced using a local search technique. The role of local search in memetic algorithms is to locate the local optimum more efficiently then the tabu search. Figure 3 explains the generic implementation of memetic algorithm.

```
Encode solution space
set pop_size, max_gen, gen=0;
set cross_rate, mutate_rate;
initialize population
while(gen < gensize)
Apply   generic GA
Apply local search
end while
Apply final local search to best
chromosome
```

Figure 3: The memetic algorithm

4.3.1 Hill climbing local search algorithm

The hill climbing search algorithm is a local search and is shown in figure 4. It is simply a loop that continuously moves in the direction of increasing quality value[15]

```
While (termination condition ins not satisfied)
do
        New sol  $\leftarrow$  neighbors(best sol );
         If new sol  is better then actual sol  then
                  Best sol $\leftarrow$ actual sol
          End if
End while
```

Figure 4 : The Hill climbing local search algorithm

5. Result & discussions

In this section a number of experiments are carried out which outlines the effectiveness of both the algorithm described above. The purpose of these experiments is to compare the performance of memetic algorithm approach with tabu search approach for the cryptanalysis of simplified SDES algorithm. The experiments were implemented in ‘C’ on a Pentium IV(1.83 Ghtz). Experimental results obtained from these algorithms were generated with 100 runs per data point e.g. ten different messages were created for both the algorithms and each algorithm was run 10 times per message. The best result for each message was averaged to produce data point.

Table 1 displays the results obtained from our memetic algorithm together with the result produced by the tabu search.

Amount of Cipher text	Memetic Algorithm			Tabu Search		
	TIME (M)	Std. dev	Number of bit matched in the key (N)	TIME (M)	std. dev	Number of bit matched in the key (N)
100	5.1	4.70	8	7.9	4.82	6.18
200	14	3.40	6	8.6	6.13	6
300	15.3	2.72	5	8.4	6.01	4
400	12.5	2.27	7	8.5	4.61	6
500	10	2.16	6	10.3	4.61	6
600	5.5	1.86	8	8.6	4.37	7
700	3.05	1.73	7	8.5	4.42	6
800	2.85	1.59	8	8.5	3.39	8
900	2.24	1.56	9	8.4	2.23	6
1000	2.14	. 1.49	9.17	8.4	2.20	8

Table 1: Comparison of memetic algorithm and Tabu search

This table basically compares the average number of key elements (out of 10) correctly recovered versus the amount of cipher text and the computation time to recover the keys from the search space. The table shows results for amounts of cipher text ranging from 100 to 1000 characters.

From figure 5, the first point to note is that the numbers of keys obtained from both the algorithms are acceptable.

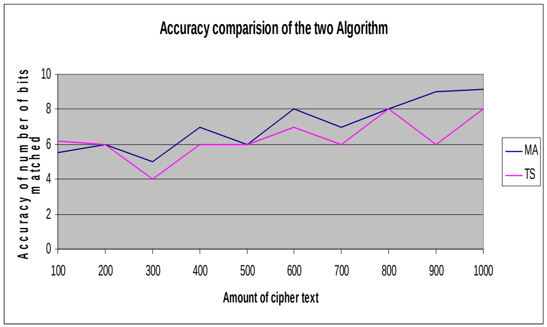

Figure 5 : The Accuracy comparison of memetic algorithm and tabu search

From Table 1, it can be seen that the standard deviation values for memetic algorithm is less than tabu search this shows that memetic algorithm has a less variance in its results. So statistically, it can be proved that the performance of memetic algorithm is superior, in term of number of bits recovered from the key.

Also we can say that including a high quality heuristic solution can help the memetic algorithm to improve its performance by reducing the likelihood of its premature convergence. Comparing the running time of the two algorithms, we found that tabu search is not sensitive to the amount of cipher text.

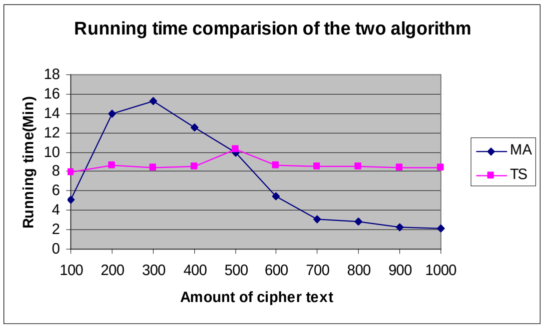

Figure 6: The running time comparison of memetic and tabu search

Figure 6 clearly shows that the running time of memetic algorithm is severely reduced as we are increasing the amount of cipher text whereas results suggest that the tabu search is unaffected.

Tabu search can be seen to be the most efficient algorithm as almost same keys is achieved in shorter time. In contrast memetic algorithm is more sensitive to amount of cipher text, for a large amount of cipher text the memetic algorithm can be seen outperform Tabu search. It means a small amount of cipher text provides an insufficient search space, which memetic algorithms perform poorly. However, a large amount of cipher text is having the large search space, possibly resulting improvement in case of memetic algorithm.

6. Conclusion

In this paper we have presented a memetic algorithm & tabu search approach for the cryptanalysis of simplified data encryption standard algorithm – A challenging optimization problem in NP-Hard combinatorial problem. A memetic algorithm is an extension of the traditional genetic algorithm. It is based on a genetic algorithm extended by a search technique to further improve individual’s fitness that may keep high population, diversity and reduce the likelihood premature convergence.

Our objective is to determine the performance of memetic algorithm in comparison with tabu search for the cryptanalysis of SDES. The first performance comparison was made on the average number of key elements (out of 10) correctly recovered versus the amount of ciphertext. Our experimental result shows that memetic algorithm is slightly superior for finding the number of keys accurately in comparison of tabu search because search technique is incorporated in memetic algorithm and the solution space is better searched. The second comparison was made upon the computation time for recovering the keys from the search space. From the extensive experiments, it was found that tabu search can be seen to be the most efficient algorithm as almost same keys is achieved in shorter time but in contrast for a large amount of cipher text the memetic algorithm can be seen outperform tabu search. Result indicates that memetic algorithm is extremely powerful technique for the cryptanalysis of SDES.

REFERENCES

[1] Davis, L. , “Handbook of Genetic algorithm”, Van Nostrand Reinhold, New York ,1991.

[2] Garg Poonam, Sherry A.M., Genetic algorithm & tabu search attack on the monoalphabetic substitution cipher, Paradime Vol IX no.1, January-June 2005,pg 106-109

[3] Garg Poonam, Shastri Aditya, Agarwal D.C, Genetic algorithm, Tabu Search & Simulated annealing Attack on Transposition Cipher ,proceeding of Third AIMS International conference on management at IIMA – 2006, pg 983-989

[4] Garg Poonam, Shastri Aditya, An improved cryptanalytic attack on knapsack cipher using Genetic algorithm approach, International journal of information technology, volume 3 number 3 2006, ISSN 1305-2403, 145-152

[5] Garg Poonam, Genetic algorithm Attack on Simplified Data Encryption Standard Algorithm, International journal Research in Computing Science, ISSN 1870-4069, 2006.

[6] Garg Poonam, Memetic Algorithm Attack on Simplified Data Encryption Standard Algorithm, proceeding of International Conference on Data Management, February 2008, pg 1097-1108 .

[7] Goldberg, D.E., “Genetic algorithm in Search, Optimization and Machine Learning”, Addison-Wesley, Reading, 1989.

[8] Giddy J. P and Safavi-Naini R., Automated cryptanalysis of transposition ciphers, The Computer Journal, Vol 37, No. 5, 1994.

[9] Glover F., Tabu search: A tutorial Interfaces, 20(4): 74-94, July 1990.

[10] Holland, J., “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, 1975.

[11] Kolodziejczyk, J., Miller, J., & Phillips, P. ,The application of Genetic algorithm in cryptoanalysis of knapsack cipher, In Krasnoproshin, V., Soldek, J., 1997

[12] Methew, R.A.J. (1993, April), The use of genetic algorithm in cryptanalysis, Cryptologia, 7(4), 187-201.

[13] Nalini, Cryptanalysis of Simplified data encryption standard via Optimization heuristics, International Journal of Computer Sciences and network security, vol 6, No 1B, Jan 2006

[14] P. Men and B. Freisleben, "Memetic Algorithms for the Traveling Salesman Problem," Complex Systems, 13(4):297-345. 2001.

[15] P, Moscato, "on evolution, search, optimization. genetic algorithm and martial arts: toward memetic olgorithms", Technical report, California, 1989.

[16] Spillman et. al., Use of a tabu search in the cryptanalysis of simple substitution ciphers, Cryptologia, 17(1):187-201, April 1993.

[17] Spillman R.,Cryptanalysis of knapsack ciphers using genetic algorithm. Cryptologia, 17(4):367–377, October 1993.

[18] Schaefer E, A simplified data encryption standard algorithm, Cryptologia, Vol 20, No 1, 77-84, 1996.

[19] Yaseen, I.F.T., & Sahasrabuddhe, H.V. (1999), A tabu search for the Cryptanalysis of Chor-Rivest knapsack public key cryptosystem (PKC), In Proceedings of Third International Conference on Computational Intelligence and Multimedia Applications, pp. 81-85, 1999