Journal of Computer Science 3 (3): 134-137, 2007
ISSN 1549-3636

Genetic Algorithm and Tabu Search Attack on the Mono-Alphabetic Subsitution Cipher in Adhoc Networks

[1]A. K. Verma, [2]Mayank Dave and [3]R. C. Joshi
[1]Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology Deemed University, Patiala, PB, India
[2]Department of Computer Engineering, National Institue of Technology, Kurukshetra, HR, India
[3]Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, UA, India

Abstract: With exponential growth of networked system and application such as e-Commerce, the demand for effective Internet security is increasing. Cryptology is the science and study of systems for secret communication. In consists of two complementary fields of study: cryptography and cryptanalysis. This study presents a cryptanalysis method based on Genetic Algorithm and Tabu Search to break a Mono-Alphabetic Substitution Cipher in Adhoc networks. We have also compared and analyzed the performance of these algorithms in automated attacks on Mono-alphabetic Substitution Cipher. The use of Tabu search is largely an unexplored area in the field of Cryptanalysis. A generalized version of these algorithms can be used for attacking other ciphers as well.

Key words: Mono-alphabetic substitution cipher, genetic algorithm, tabu search, key search

INTRODUCTION

The demand for effective network security is increasing exponentially day by day. Businesses have an obligation to protect sensitive data from loss or theft. Not only businesses see to the security needs; they have to understand where the computer is vulnerable and how to protect it. In the present scenario, where a user needs to be connected *anyhow, anywhere, anytime* we use ad-hoc networks[1,2]. Ad Hoc Networks are highly vulnerable to security attacks so there is a need to develop a scheme to guarantee certain properties of information (availability, confidentiality, authenticity, integrity). Cryptology is at the heart of providing such guarantee. Cryptology is concerned with the making (Cryptography) and breaking (Cryptanalysis) of Scheme. Cryptography applied by authorized information sharers to design and develop encryption scheme in order to ensure confidentiality of information while cryptanalysis method uses mathematical and statistical attempts by unauthorized person to break ciphers in order to reveal the meaning of the underlying protected data. The cryptanalyst looks for trapdoors; exploitable regularities in either the cipher system or the language or both, combinations of plaintext and ciphertext; or anything else which may prove helpful in breaking the cipher.

Classical ciphers fall into one of the two broad categories: Substitution cipher & transposition cipher. Modern cryptosystems have now supplanted the classical ciphers but cryptanalysis of classical ciphers is most popular crypto-logical application for meta-heuristic search research. The basic concepts of substitution and transposition are still widely used today in Advanced Encryption Standard (AES). International Data Encryption Algorithm (IDEA) is also widely used encryption algorithm, which uses only three very simple operators, namely substitution, permutation (transposition) and bit-wise exclusive-OR operator. Since the operations of the classical cipher are the building blocks of modern ciphers, so the classical ciphers are usually the first ones considered when researching new attacks.

In a mono-alphabetic substitution cipher the value of a character or character string is changed when transforming the plaintext into ciphertext, but the position of the original string and its value replacement correspond exactly in the plain and ciphertext. This kind of scheme is usually found in the recreational crypto columns of popular publications such as newspaper and magazines.

For example, if we encrypt the plaintext “how are you” using a single character Mono-alphabetic Substitution Cipher with the shown in Fig. 1 the cipher text is ABSHLCZBX is obtained. In this case the key space consists of all possible permutation tables of the form in Fig. 1 the size of this key space 26! = 403 291 461 126 605 635 584 000 000, which clearly preempts a brute force search in order to break the system.

Although this cipher is no more secure, it turns out to be surprisingly time consuming to break by hand and is an example of what is known in general as a mono-alphabetic substitution, because any single character in

Corresponding Author: A K Verma, Department of Computer Science and Engineering, Thapar Institute of Engineering & Technology, Deemed University, Patiala, PB, India

Plaintext Chararacters ->											
a b c d e f g h i j k l m n											
Ciphertext ->											
H W U G	C	T	V	A	E	K	D Y Q P				
Plaintext ->											
o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext ->											
B	R	J	L	F	I	X	M	S	O	Z	N

Fig. 1: Example of a key of a single character mono-alphabetic substitution cipher

the plaintext is mapped onto some fixed single ciphertext character every time occurs. In more complicated mono-alphabetic substitution cipher it may happen that a specific plaintext character is mapped onto any one of a variety of different ciphertext character, depending on circumstances controlled by the cipher key. Examples of other substitution ciphers include the well-known Ceasar cipher, Affine substitutions, Vigenere cipher and Beaufort cipher.

Forsyth and Safavi-Naini[3] have published an attack on the simple substitution cipher using simulated annealing and Spillman *et al.*[4] presented an attack using genetic algorithm. Dimovski and Gligoroski[5] presented an attack on poly-alphabetic substitution cipher that utilized the parallel genetic algorithm.

Gester[6] has published an attack on Substitution Ciphers using Genetic Algorithm. Jakobsen and Knudsen[7] presented an attack on block ciphers of low algebraic degree.

This study introduces an attack on the mono-alphabetic substitution cipher using the Tabu search[8]. The previously published attacks using genetic algorithm were enhanced and modified in order that an accurate comparison of two techniques could be obtained.

All experiments presented in the study were performed on text using 27 characters alphabet, i.e., A-Z and the space character. All punctuation and structure (sentences/ paragraphs) has been removed from the text before encryption. Any two words are separated by a single space character.

Genetic algorithm: The genetic algorithm is based upon Darwinian evolution theory. The genetic algorithm is modeled on a relatively simple interpretation of the evolutionary process (Fig. 2); however, it has proven to a reliable and powerful optimization technique in a wide variety of applications[9]. Holland[10] in 1975 was first to propose the use of genetic algorithms for the problem solving. Goldberg[11] was also a pioneer in the area of applying genetic processes to optimization. Over the past twenty years numerous application and adaptation of genetic algorithms have appeared in the literature.

During each iteration of the algorithm the processes of selection, reproduction and mutation each take place in order to produce the next generation of solution. The actual method used to perform each of these operations is very much dependent upon the problem being solved and the representation of the solution.

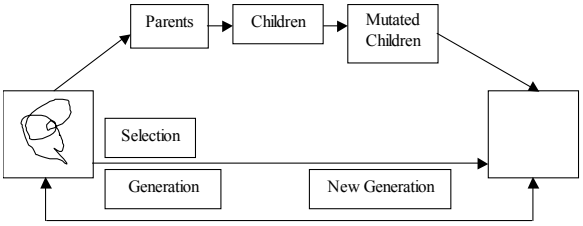

Fig. 2: The evolutionary process

Tabu search: Glover[8] was pioneer in use of the Tabu search and 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, routing etc.

The properties of Tabu search can be used to enhance other procedure by preventing them becoming stuck in the regions of local minima (Fig. 3). The Tabu search, like the genetic algorithm, introduces memory structures into its workings. In this case, the purpose of the memory is multi-faceted. The genetic algorithm utilizes its solution pool as a mechanism for introducing diversity into breeding process. The Tabu search utilizes memory for a additional purpose, namely to prevent the search from returning to a previously explored regions 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 override the Tabu restrictions making it possible to broaden the search for the global optimum.

Much of the implementation of the Tabu search is problem specific, i.e., the mechanisms used depend heavily upon the type of problem being solved.

1. Initialize algorithm variable: G the maximum number of generations to consider, M the solution pool size and any other problem dependent variable.
2. Generate an initial solution pool containing M candidate solution. This initial pool can be generated randomly or by using a simple known heuristic for generating solutions to the problem in hand. This solution pool is now referred to as the current solution pool.
3. For G iterations, using the current pool:
   a. Select a breeding pool from the current solution pool from the current solution pool and make pairing of parents.
   b. For each parental pairing, generate a pair of children using a suitable mating function.
   c. Apply a mutation operation to each of the newly created children.
   d. Evaluate the fitness function of each of the children.
   e. Based on the fitness of each of the children and the fitness of each of the solutions in the current pool, decide which solution will be placed in the new solution pool. Copy the chosen solutions into the new solution pool.
   f. Replace the current solution pool with new one. So, the new solution pool becomes the current one.
4. Choose the fittest solution of the final generation as the best solution.

Fig. 3: The genetic algorithm

An initial solution is generated (usually randomly). The Tabu list is initiated 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 solutions is proposed in every iteration. The most admissible 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.

MATERIALS AND METHODS

The technique used 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], where _ represents the space symbol, 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 assignjing of different weights to each of the three n-gram types.

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

Spillman *et al.*[4], in their attack on the simple substitution cipher uses equation 1. This equation is based on unigram and bigram statistics.

$$fitness = (1 - \sum_{i=1}^{26} \{|SF[i] - DF[i]| + \sum_{i=1}^{26} |SDF[i,j] - DDF[I,Jj]|\}/4)^8 \quad (2)$$

SF [i] is the standard frequency of character i in the English plaintext, while DF [i] is the measured frequency of the decoded character in the ciphertext the function.

Jakobsen and Knudsen[7] in his attack uses a

$$C_k = \sum_{i,j \in A} |K^b_{(i,j)} - D^b_{(i,j)}| \quad (3)$$

The only diffrerence between these assessment functions is the inclusion of different statistics (Equations 3 and 1 are equal if $\alpha = \gamma = 0$). The complexity of equation 1 is $O(n^3)$, where n is the general alphabet size, when trigram statistics are used. The effectiveness of using the different n-grams is evaluated using a range of weights and is concluded that trigram are the most effective basis for a cost function, but the benefit of using trigram over bigram is small. In fact, due to the added complexity of using trigram, it is usually more practical to use only unigram and bigram[12].

RESULTS

Experimental results for the two algorithms are generated with 300 runs per data point. Each of the attack was run a number of times with a variety of parameter values. Each algorithm was run with initial keys being chosen randomly.

We have considered two metrics for making a comparison. The first metric is made upon the amount of cipher text available to attack. These results are presented in Table 1. The result in Table 1 represents the average number of key elements correctly placed for a key size of 27. Due to limited length of ciphertext none of the key is true key. Still a large portion of the cipher text be decrypted correctly, the message was almost readable.

Table 1: The amount of key recovered versus available

Amount of Ciphertext	GA	TS
200	13.17	11.75
400	20.72	18.40
600	22.59	22.40
800	23.30	22.40
1000	23.59	24.18

Table 2: Time Comparison of Methods

Algorithm	Average Time
GA	240
TS	95

The second metric is made upon the time taken by the algorithms. Table 2 gives an indication of the convergence rates of each of the algorithm as a function of the number of iterations.

The genetic algorithm appears to be slow, although the convergence rate improves as gene pool collects more fit solution. The result in shown in Fig. 4 illustrates that the Tabu search requires much less

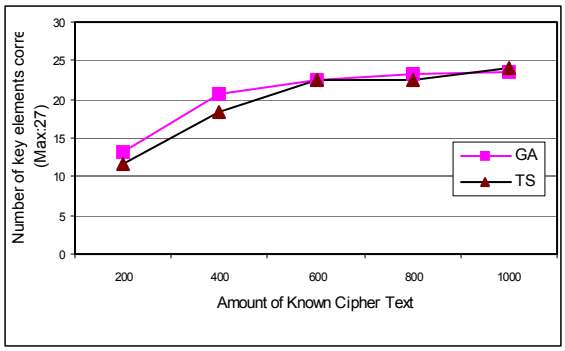

Fig. 4:

iteration to find the correct solution and the algorithm converged to a solution very fast.

CONCLUSION

This article presented Genetic Algorithm and Tabu search attack on the mono-alphabetic substitution cipher. In this we have considered a number of automated attacks against mono-alphabetic substitution ciphers. The principles used in mono-alphabetic substitution ciphers form the foundation for many of the modern cryptosystems. The first performance comparison was made on the average number of key elements (out of 27) correctly recorded versus the amount of ciphertext, which is assumed to be known in the attack. It was found that for mono-alphabetic substitution cipher both the algorithms performed equally with respect to amount of known cipher text available to attack. The second comparison was made upon the time taken by the algorithms it was found that the Tabu search required less time to find the correct solution. Results indicate that Tabu search is extremely powerful technique for attack on mono-alphabetic substitution cipher.

REFERENCES

1. Perkins, C.E., 2001. Ad hoc Networking. Addison-Wesley.
2. Toh, C.K., 2002. Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall PTR.
3. Forsyth, W.S. and R. Safavi-Naini, 1993. Automated cryptanalysis of substitution ciphers. Cryptologia, 17: 407-418.
4. Spillman, R. *et al*., 193. Algorithm in the cryptanalysis of simple substitution ciphers.
5. Dimovski, D.G., 2003. Alphabetic substitution cipher using a parallel genetic algorithm domain cooperation through SCOPES PROJECT. Ohrid, Maccedonia.
6. Gester, J., 2006. Solving substitution ciphers with genetics algorithm. http://www.cs.rochester.edu/u/brown/Crypto/studprojs/SubstGen.pdf, June 2006.
7. Jakobsen, T. and L.R. Knudsen, 2001. Attacks on block ciphers of low algebraic degree. J. Cryptology, 14: 197-210.
8. Glover, F. and M. Laguna, 1997. Tabu Search. Kluwer Academic Publishers, Boston.
9. Davis, L., 1991. Handbook of Genetic Algorithms. Van Nostrand Reinhold.
10. Holland, J.H., 1975. Adpatation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI.
11. Goldberg, D.E., 1991. Genetic Algorithm in Search, Optimization and Machine Learning. Addison Wesley.
12. Wagner, S., M. Affenzeller and D. Schragl, 2004. Traps and Dangers when Modelling Problems for Genetic Algorithms. Cybernetics and Systems.