An Artificial Life Technique for the Cryptanalysis of Simple Substitution Ciphers

Mohammad Faisal Uddin[1], Amr M. Youssef[2]

Department of Electrical and Computer Engineering[1]
Concordia Institute for Information Systems Engineering[2]
Concordia University, Montreal, Quebec, Canada
mf_uddin@encs.concordia.ca, youssef@ciise.concordia.ca

ABSTRACT- In this paper, we investigate the use of Ant Colony Optimization (ACO) for automated cryptanalysis of classical simple substitution ciphers. Based on our experiments, ACO-based attacks proved to be very effective on various sets of encoding keys.

Keywords: *Cryptanalysis, Simple Substitution Cipher, Ant Colony Optimization.*

I. INTRODUCTION

Cryptanalysis, from the Greek *kryptós*, "hidden", and *analýein*, "to loosen", is the art and science of breaking, i.e., decoding ciphertext into its corresponding plaintext, without prior knowledge of the secret key. In general, classical ciphers operate on an alphabet of letters (e.g., "A-Z"), and can be implemented by hand or with simple mechanical devices [1]. Classical ciphers are often divided into transposition ciphers and substitution ciphers. In a substitution cipher, letters are systematically replaced throughout the message for other letters. As far as security is concerned, classical ciphers are no match for today's ciphers. However, in principles, they have not lost their significance because most of the commonly used modern ciphers utilize the operation of classical cipher as their building blocks. In fact, most complex algorithms are formed by mixing substitution and transposition in a product cipher. Modern block ciphers such as DES and AES iterate through several stages of substitution and transposition.

Ant Colony Optimization (ACO) [2], a recently proposed population based optimization technique inspired the behavior of real ant colonies, has been used to solve different discrete optimization problems. In this paper, we investigate the use of ACO in automated cryptanalysis of classical substitution ciphers. Based on our experiments, ACO-based cryptanalysis proved to be very effective on various sets of encoding keys.

The rest of the paper is organized as follows. In section II, we briefly review some of the previous work related to cryptanalysis of classical ciphers. In section III, we outline the simple substitution cipher. Section IV summarizes the n-gram statistics and the cost function used in our attack. In Section V, we review the ACO algorithm and describe how it is adopted for the cryptanalysis of substitution ciphers. In section VI, we explain our experimental setup and report the obtained results. Finally, section VII is the conclusion.

II. PREVIOUS WORKS

The Arabs were the first to make significant advances in cryptanalysis. Early in the 15th century, an Arabic author, Qalqashandi, wrote down a technique for solving ciphers using the average frequency of each letter of the language [1].

Over the last twenty-five years, several optimization heuristics have shown promise for automated cryptanalysis of classical ciphers [3]. One of the early proposals was given by Peleg and Rosenfeld [4]. They modeld the problem of breaking substitution ciphers as a probabilistic labeling problem. Every coded alphabet was assigned probabilities of representing plaintext alphabets. These probabilities were updated using the joint letters. Using this scheme in an iterative way they were able to break the cipher. Carrol and Martin [5] developed an expert system approach to solve simple substitution ciphers using hand-coded heuristics. Forsyth and Safavi-Naini [6] recast the problem as a combinatorial optimization problem and presented an attack on simple substitution cipher using simulated annealing algorithm. Spillman *et. al* [7] presented an attack on simple substitution cipher using genetic algorithm. Clerk [3] re-implemented the genetic algorithm and simulated annealing attack in order to compare them and also evaluate a third technique using *tabu* search. Bahler and King [8] used trigram statistics and relaxation scheme to iterate towards the most probable key as previously done by Peleg and Rosenfeld. Lucks [9] used a word pattern dictionary and search over it with the constraint that all ciphertext characters must decrypt to the same plaintext character. Hart [10] improved upon this method by directing this combinatorial search towards more frequent English

1-4244-0038-4 2006
IEEE CCECE/CCGEI, Ottawa, May 2006

words. Recently, Ant Colony Optimization (ACO) was used successfully in breaking transposition ciphers [11].

III. SIMPLE SUBSTITUTION CIPHER

Simple substitution cipher is a well-known cryptosystem. It is the simplest form of substitution ciphers. Each symbol in the plaintext maps to a different symbol in the ciphertext [1]. The simple substitution cipher used in this work operates on the English alphabet of 26 letters ("A-Z"). We assume that all the punctuations and structure (sentences/paragraphs, space characters, and newline characters) are removed from the plaintext in order to hide these obvious statistics from the ciphertext.

Key:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
X N Y A H P O G Z Q W B T S F L R C V M U E K J D I
Encryption:
Plaintext:
ANTCOLONYOPTIMIZATIONISAPOWERFULTOOL
Ciphertext:
XSMYFBFSDFLMZTZIXMZFSZVXLFKHCPUBMFFB

Table 1: Example of a Simple Substitution Cipher

Let x be an n-character alphabet $\{x_0,x_1,x_2,x_3,\ldots,x_{n-1}\}$, and y is also an n-character alphabet $\{K(x_0), K(x_1), K(x_2), K(x_3),\ldots, K(x_{n-1})\}$, where K: $x \rightarrow y$ is a one to one mapping of every alphabet of x to the corresponding alphabets of y. Here 'K' is the cipher key function which can be looked at as a permutation of the 26 character. The transmitter enciphers the plaintext into ciphertext with a predetermined key function (K) and sends it to the receiver. The receiver deciphers the ciphertext to plaintext with the inverse key function (K^{-1}).
An example for a simple substitution cipher key and encryption operation is shown in Table 1.

IV. N-GRAM STATISTICS AND COST FUNCTION

There exists $26! \approx 4.03291461 \times 10^{26} \approx 2^{88}$ possible keys for a simple substitution cipher with alphabet size of 26 characters. This number is obviously too large to allow any kind of exhaustive search.
However, one special property of simple substitution cipher that makes it relatively easy to cryptanalyze, is that the language statistics remain unchanged by the encryption process and hence frequency analysis presents a basic tool for breaking classical ciphers. In natural languages, certain letters of the alphabet appear more frequently than others. The n-gram statistics indicates the frequency distribution of all possible instances of n adjacent characters [1]. For example 'E' is the most common unigram (1-gram) in English language, and one of the common bigram (2-gram) in English language is 'TH'. These n-gram statistics can be used to measure the fitness of a suggested decryption key.
In our case we have only considered unigram and bigram statistics for deciphering the ciphertext. These statistics can be easily calculated from any large English corpus.

For our experiments, these statistics were generated from an online version of the book "Twenty Thousand Leagues Under the Sea" by Jules Verne.

Let R^U, R^B and DK^U, DK^B be the reference language unigram and bigram statistics and decrypted message unigram and bigram statistics (using a key K) respectively. Then, our cryptanalysis problem corresponds to finding a decryption key, K, that minimizes the following weighted objective function

$$Cost(K) = \lambda_1 \sum_{c \in \{A,B,\ldots,Z\}} \left| R^U_{(c)} - DK^U_{(c)} \right| + \lambda_2 \sum_{c_1,c_2 \in \{A,B,\ldots,Z\}} \left| R^B_{(c_1,c_2)} - DK^B_{(c_1,c_2)} \right|$$

V. ANT COLONY OPTIMIZATION (ACO)

Ant Colony Optimization [2] is a heuristic optimization method for solving different combinatorial optimization problems. It is a population based approach which borrows ideas from biological ants. The social behaviors of ants have been much studied by the scientists and from their behavior the computer scientists came up with the idea of this optimization. Experiments with real ants showed that ants go from the nest to the food source and backwards, then after a while, the ants prefer the shortest path from the nest to the food source. Real ants are capable of finding the shortest path from their nest to a food source without using visual cue [12]. Ants have a special way of communicating information concerning food sources. While walking, ants secrete an aromatic essence on the ground, called pheromone. The other ants will follow the path of greater pheromone trail with higher probability and as they will follow the path they as well will secrete pheromone there. So the pheromone of that path which greater number of ants are following will increase and as the pheromone trail of that path increases more ants will follow that path. Since ants passing through food source by shorter path will come back to the nest sooner than ants passing through longer paths, the shorter path will have a higher traffic density than that of the longer one. Thus a single ant will follow the shortest path with higher probability [2].
Dorigo and Gambardella [13] presented ant colony system (ACS) for solving traveling salesman problem (TSP). For the TSP, the Euclidean distance between two cities, $d(i,j)$, is used to represent the *a priori* knowledge

of desirability of choice of a particular city while the ants are in some particular city. For our cryptanalysis problem each complete path constructed by ants is a permutation of the nodes (alphabetic characters) corresponding to a key. So in our algorithm, we use the distance between the unigram frequency of the reference language statistics and the target test key to represent the *a priori* desirability of choice of a particular key element, i.e., we set $d(i,j)=\left|R^U_{(i)}-DK^U_{(j)}\right|$.

In what follows we describe how we adopted the ACO to our cryptanalysis problem.

The system is initialized with a group of ants moving across a fully connected bidirectional graph of 26 nodes $(n_1,n_2,...,n_{26})$. A *tabu* list is maintained to prevent any ant from visiting the same node twice. Every possible decryption key $K=(k_1,k_2,...,k_{26})$ corresponds to a unique path along this graph $(n_1\rightarrow n_{k_1},n_2\rightarrow n_{k_2},...,n_{26}\rightarrow n_{k_{26}})$.

The algorithm proceeds by iterating through the following three basic steps:

1. Construct a solution for all ants: At each node, each ant has to make a (statistical) decision regarding the next node to visit. At the first iteration, all the ants will move randomly. However, on subsequent iterations, the ants' choices will be influenced by the intensity of the pheromone trails left by preceding ants during previous iterations. A higher level of pheromone on a given path gives an ant a stronger stimulus and thus a higher probability to follow this path. In particular, at node *i*, the ant expand its tour to node *j* with probability *p* that is given by:

$$p(i\rightarrow j)=\begin{cases}0, & node\ j\ already\ visited\\ \dfrac{\tau(i,j)^{\alpha}\,d(i,j)^{-\beta}}{\sum_{k\ not\ visited}\tau(i,k)^{\alpha}\,d(i,k)^{-\beta}}, & otherwise.\end{cases}$$

Setting $\alpha=0$ in the above equation corresponds to the system that relies only on the unigram statistics for the cryptanalysis. For the bigram based system, the optimum value of α and β is found by a heuristic trial and error.

2. Do a global pheromone update: Once the tour is completed, every ant updates the pheromone $\tau(i,j)$ over the arc $(i-\ j)$ along its visited path as follows:

$$\tau(i,j)=\tau(i,j)+\Delta\tau(i,j)$$

where

$$\Delta\tau(i,j)=1/Cost(K)$$

3. Evaporate pheromone: After each iteration, a portion of the pheromone of the edge is evaporated according to a local updating rule,

$$\tau(i,j)=\rho\times\tau(i,j),\qquad \rho<1,$$

such that the probability of the selection of that edge by other ants decreases. This prevents construction of similar paths by the set of ants and increases the diversity of the system. The rate of evaporation provides a compromise between the rate of convergence and reliability of convergence. Fast evaporation causes the search algorithm to be stuck at local optima, while slow evaporation lowers the rate of convergence. After enough iteration of the algorithm, the pheromone of the good edges which are used in constructing of low-cost paths will increase and the pheromone of the other edges will evaporate. Thus, in the higher iterations the probability of constructing low-cost paths increases.

VI. EXPERIMENTAL RESULTS

Throughout all of our experiments, the number of ANTS was set to 1000. The rest of the parameters were varied in an ad-hoc way to optimize the results.

Figure 1 shows how the average (over 100 randomly selected keys) number of corrected key elements varies with the amount of known ciphertext. Similarly, Figure 2 shows the percentage of corrected characters versus the amount of known ciphertext. Figure 3 shows the error distribution for 100 randomly selected keys when the amount of known ciphertext is 900 characters. For this case, the average and variance of the number of errors in the recovered key characters are 1.72 and 2.9303 respectively.

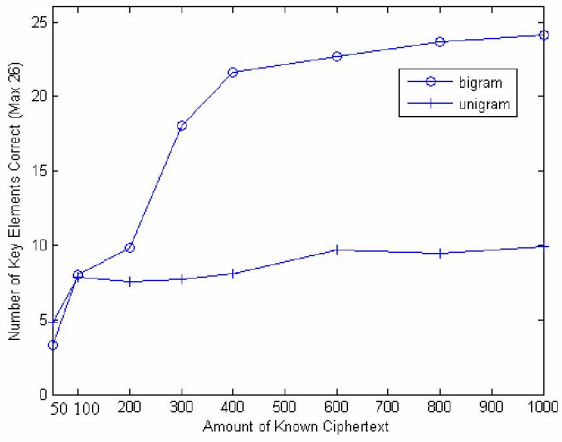

Figure 1: Number of corrected key elements versus the amount of known ciphertext

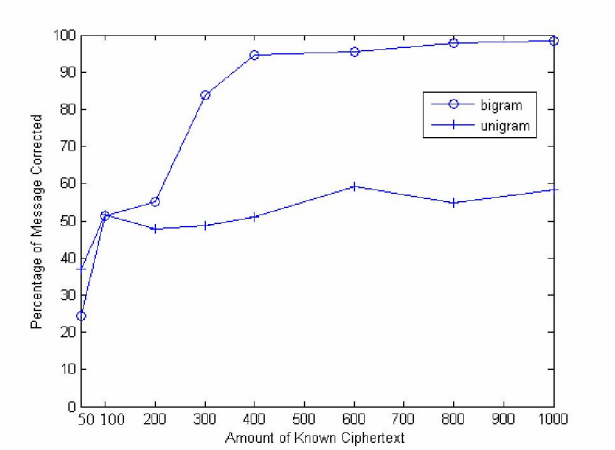

Figure 2: Percentage of corrected characters versus the amount of known ciphertext

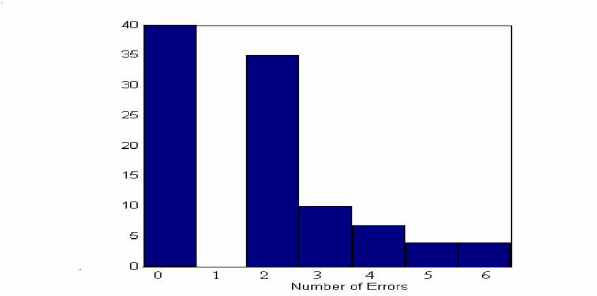

Figure 3: Error distribution for bigram based system (100 random keys and 900 known ciphertext characters)

VII. CONCLUSIONS

ACO provides a very powerful tool for the cryptanalysis of simple substitution ciphers using a ciphertext only attack. Given the noticeable accuracy gain of the bigram based attack as compared to the unigram based one; it is interesting to try the trigram in the evaluation function. We only investigated the two cases corresponding to $(\lambda_1, \lambda_2) = (1,0)$ and $(\lambda_1, \lambda_2) = (0,1)$. One may also try to use a cost function that is based on a weight combination of the different n-gram statistics.

One main disadvantage of heuristic optimization techniques (including ACO) is its large sensitivity to parameter variations (e.g., ρ, α, and β in ACO). Although fine tuning of these parameters can be done by trial and error, it is interesting to find analytical formula for the optimal (regions) of these parameters.

REFERENCES

[1] D. Kahn, "The Code breakers: the story of secret writing," revised edition, 1996.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a Colony of Cooperating Agents," *IEEE Transactions on Systems, Man, and Cybernetics. B*, vol.26, no.2, pp. 29–41, 1996.

[3] A. Clerk, "Optimisation Heuristics for Cryptology," PhD thesis, Queensland University of Technology, 1998.

[4] S. Peleg and A. Rosenfeld, "Breaking substitution ciphers using a relaxation algorithm," *Communications of the ACM*, vol. 22(11), pp.598–605, 1979.

[5] J. Carrol and S. Martin, "The automated cryptanalysis of substitution ciphers," *Cryptologia,* vol.10(4), pp.193–209, 1986.

[6] W. S. Forsyth and R. Safavi-Naini, "Automated cryptanalysis of substitution ciphers," *Cryptologia*, vol.17(4), pp.407–418, 1993.

[7] R. Spillman, M. Janssen, B. Nelson and M. Kepner, "Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers," *Cryptologia,* vol.17(1), pp.31–44, 1993.

[8] D. Bahler and J. King, "An implementation of probabilistic relaxation in the cryptanalysis of simple substitution systems," *Cryptologia,* vol.16(3), pp.219–225, 1992.

[9] M. Lucks, "A constraint satisfaction algorithm for the automated decryption of simple substitution Ciphers," *In Proceedings of CRYPTO'88*, pp. 132–144, 1988.

[10] G. W. Hart, "To decode short cryptograms," *Communications of the ACM,* vol.37(9), pp.102–108, 1994.

[11] M.D. Russell, J.A. Clark, and S. Stepney, "Making the most of two heuristics: breaking transposition ciphers with ants," *The Congress on Evolutionary Computation (CEC '03),* Vol. 4, pp.2653- 2658 ,2003.

[12] R. Beckers, J.L. Deneubourg and S. Goss, "Trails and U-turns in the selection of the shortest path by the ant Lasius Niger," *Journal of Theoretical Biology,* vol.159, pp.397–415, 1992.

[13] M. Dorigo, L.M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation, vol. 1, no.1, pp.53–66, 1997.