ISSN 1512-1232

Using Genetic Algorithm for the Breaking Vigenere Cipher

Zurab Kochladze, Giorgi Gelashvili

Ivane Javakhishvili Tbilisi State University
Faculty of Exact and Natural Sciences, Department of Computer Science
0186, University street 13. Tbilisi, Georgia, zurab.kochladze@tsu.ge

Abstract

This paper discusses the use of genetic algorithms to cryptanalysis. Namely, how to break up the Vigenere cipher with the genetic algorithm and why it is impossible to break complete the cipher with this method today.

Keywords: *Vigenere cipher, genetic algorithms, natural language analysis.*

I. Introduction

Alongside of modern information technologies development, it becomes increasingly important to prevent the unauthorized access to information stored in computer systems and transmitted through communication channels. The critical role for problem solving belongs to the cryptology that develops very quickly. The creation of new cryptographic systems for security and protection of information, naturally, requires the development of cryptanalysis methods to identify the areas of application of these systems, suitability and resilience to various cryptographic attacks.

In general, the goal of cryptanalysis is to break the cryptographic algorithm, or to find the key of this algorithm, or create an algorithm that will be able without key to restore the plaintext from the ciphertext.

As it is known, the resilience of cryptographic algorithms to attacks significantly depended on the space of the keys. The larger the space, the more difficult is to break the algorithm. Practically, during cryptanalysis, the task is to build the algorithms that will reduce search area by using different mathematical or heuristic functions and will find the keys in polynomial time. In recent years, the genetic algorithms are often used for this purpose [1, 2, 3, 4, 5] as they have greater advantages in comparison with heuristic algorithms.

Nowadays the classical ciphers exhausted its possibilities and they are not used. However, there are still some tasks that may have no practical application, because there exist other methods that can break algorithms just by attacking based only on the ciphertext**.** But despite that, they are still interesting due to their content. The fact, that genetic algorithms are often used to break classical ciphers [2, 3, 5, 6], proves that.

The present article discusses one of this kind of task, do it possible or not that genetic algorithm breaks the Vigenere cipher independently, without human intervention. The existing classic method of attacking this cipher, the Kassiski test, allows us to find the length of the key and reduce the assignment to the frequency analysis task. Itself the use of frequency analysis requires not only the knowledge of the frequencies of the letters usage in the given natural language, but also the knowledge of the morphology and grammar of the language.

II. The Vigenere cipher.

Vigenere cipher was created in 1553 by Italian cryptologist Giovanni Battista Belazzo [7] and represents polyalphabetic substitution cipher. In contrast to simple substitution ciphers, in polyalphabetic ciphers the same symbol of the plaintext in the ciphertext corresponds to different symbols, therefore structure of the plaintext does not transform one to one in the ciphertext. Consequently, it is impossible to break this cipher through use of attack cryptogram based on frequency analysis method.

ISSN 1512-1232

Let's describe the principle of the algorithm. Let's assume that the alphabet, which consists from n amount of symbols, that can be represented as plaintext, as well as key and ciphertext (as in the alphabets of the natural languages plus punctuation marks). Then as plaintext, as well as key and ciphertext can be presented as the strings drawn up by these symbols. Let's note the sequence of plaintext $m_1, m_2, ..., m_l$ symbols. Therefore $k_1, k_2, ..., k_i$ will be a sequence of symbols of the key and $c_1, c_2, ..., c_i$ will be sequence of cryptographic symbols. As a rule, the length of the key in this algorithm is smaller compared to the plaintext. Therefore, the encryption undergoes by blocks. The plaintext should be divided into blocks that corresponds the key length. Each block should be encrypted separately from each other. Let's reconcile alphabet characters to numbers from 0 till $(n-1)$, where n is the number of characters in alphabet. Then we can transform the plain text as well as the key letters in the numbers. The encryption formula in this case looks as

$$c_i = m_i + k_i (\mathrm{mod}\, n).$$

After the encryption of each block, let's unite the blocks again. The received numerical sequence should be transformed into the symbols. The encrypted text is ready.

For almost three hundred years there was no algorithm for breaking this cipher alone on the basis of ciphertext attack. Only English mathematician Charles Bebbage in the nineteenth century, and later Mr. Kassiski, the retired Major of Prussian army, establish the method that enables to attack the cipher only based on encrypted text. This method in the history of cryptography known as the Kassiski test. According to the procedure, it will find the same combination of the characters (like bigrams, trigrams and etc.), will calculate distance between them (the number of symbols) and will identify largest common division of these numbers. This division will be the length of the key. After the ciphertext should be divided into blocks of this length and write in rows to get a matrix. Each column of the given matrix will be encrypted with the same key, which gives us the opportunity to use the frequency analysis method to each column.

The famous American cryptanalyst in 1920 W. Friedman introduced the statistical characteristics of the match, which allows determine the length of the key and itself key much easier.

III. The Task

Our goal was to break the Vigenere cipher using a genetic algorithm. For this purpose, we created the algorithm program and encrypted several different Georgian texts. We used only the thirty-three letters of the alphabet (without punctuation marks). In addition, the words in the cryptogram were separated from each other.

The genetic algorithms.

The genetic algorithms represent one of the most widely used areas of evolutionary programming [, 8, 9]. The genetic algorithms act in accordance with the principles of natural evolution. Genetic algorithms represent Iterative algorithms that replicate the same operations for different possible solutions until they find the best solution or do not exhaust resources.

In order to build a genetic algorithm, it is necessary to define a set of possible solutions and the form they will be represented in the computer. In our case, it was the sequence of the keys from thirty-three letters of the Georgian alphabet.

Defining the initial set of possible solutions of the given task, i.e. the population. Determine the number of possible solutions of this population.

Describe the genetic operations that are used to get new possible solutions from the possible solutions of population.

Determine the compatibility function, e.g. fitness function, to evaluate those possible solutions of population and determine the principle for the reorganization of the population. Determine the stoppage moment for the algorithm.

The genetic algorithm created by us missed the process of determining the length of the key, as it was not issue of the interest. Therefore, knowing the length of the key, 20 initial solutions of

ISSN 1512-1232
the given length was chosen that consist of randomly taken letters. The number of iterations determined as 50. The Genetic operators chosen are one and two-point crossovers and mutation. The fitness function explained as follows - algorithm calculates the number of symbols in the ciphertext and subtracts the number of symbols identified by the given possible solution. Moreover, if the word is fully understood in the text, given possible solution gets one point on each word. The algorithm stops when the fitness-function becomes zero, or when the number of iterations reaches a certain number (in our case, fifty).

The software is designed in C#. The algorithm uses the database **MSSQL Server**. The base name is **Alphabet** and contains three tables: **Bigrams, Trigrams** and **Geotext.**

The Tables **Bigrams and Trigrams** include all bigrams and trigrams that can be formed from 33 letters. The frequencies of their usage in texts are indicated. Table **Geotext** contains 300000 words with indication of the frequency of use.

The algorithm starts work with initial possible solution. They will decrypt ciphertext. For identification of decrypted text, it starts searching in **Geotext.** During the search of words the following principle applied: If the same part of the decrypted text can be two words, the preference is given a word that contains more letters. For instance, if the word "ხე" (tree) and "ხელი" (hand) comes from one and the same decrypted section, the algorithm chooses the word "ხელი“ (hand). Then it will move on to examine the trigrams and bags. After completing this procedure, we have full or incomplete texts with each possible solution.

If the text fully encrypted and the plaintext text is restored, the algorithm stops work. In opposite case, the algorithm starts assessment of the possible solution through fitness function. It subtracts from the amount of the encrypted texts symbols the sum of the words, trigrams and bigrams letters encrypted by given possible solution, and adds the amount of fully recognized words. After that algorithm starts the organizing of possible solutions list based on assessment score and switches to the genetic operations. If the possible solution recognizes the word, trigram or bigram, the initial and last points of recognized part represent the points of chromosome disruption and two or one point crossover operations take place. With this method, from 25% of existing solution candidates the new possible solutions arrive. The mutation operation is carried quite rarely (up to 5%).

The selection of the possible solution for further iteration based on following principle: all recombined possible solution remain in the list, while free places filled sporadically by old generation the possible solution, without taking into account their fitness function.

IV. Conclusion.

The experiments carried out, showed that our results are the same as in references [1,2,5,6]. In most cases the algorithm decrypts the text that is easy for human to read, however algorithm could not absolutely correctly decrypt the text. Especially hard, is for algorithm, when the words from the dictionary, like nouns change their forms and cases and getting different endings or verbs change by persons and numbers (because in dictionary the words are not given in all forms). Therefore, after the crossover operation the algorithm lose the sense of the word endings and receives absolutely the wrong team of letters. If we will interfere in the work of algorithm and in this particular case crossover operation point will put correct word endings the algorithm will gave better results. However, it turns out that the decryption happened with the human participation

These results are not random, or the fault of improperly constructed algorithm. It proves once again the without solving of problem of natural language analysis by the artificial intelligent system's such ciphers can’t be broken without human intervention. This is best shown in those classical methods of attacks that humans use while decryption. For example, the algorithm of breaking simple substation ciphers, given by Arabic cryptologists in VI- VIII centuries: ”When letter frequencies in alphabet are nearly the same, cryptanalytic must use all his/her language knowledge and after the analysis choose the right version“. This is the very knowledge that modern genetic algorithms, that attack classical ciphers, lack. Due to that, there is nothing unusual that it’s

ISSN 1512-1232
simpler for them to break modern computer ciphers [3,7,10], in which the natural language structure is almost fully excepted.

If recalled, in old ciphers it was common to rid of one or few letters, as without them was not difficult for people to understand the text. In these types of cases it's envisaged that the given algorithm practically could not decrypt the text.

Therefore, we can conclude that cryptanalysis of such ciphers without human intervention is not possible today.

References:

1. A. Dureha, A.Kaur A Generic Genetic Algorithm to Automate an Attack on Classical ciphers. International journal of Computer application (0975-8887) Vol. 64 –No.12 (February 20013).
2. Spillman R, Janssen M, Nelson B and Kepner N, Use of Genetic Algorithm in Cryptanalysis of Simple Substituion Cipher Cryptologia, Vol.17, No.4, pp. 367-377, 1993.
3. SpillmanR, Cryptanalysis of Knapsack Ciphers using Genetic Algorithms, Cryptologia, Vol.17, No.4, pp. 367-377, 1993.
4. Z. Kochladze, L. Beselia Cracking of the Merkle–Hellman Cryptosystem Using Genetic Algorithms. Transaction on Science and Technology, No3 (1-2), pp. 291-296, 2016.
5. J. Luthra, S.K.Pal A Hybrid Firefly Algorithm Using Genetic Operators for the Cryptanalysis of a Monoalphabetic Substitution Cipher", IEEE, 2011.
6. A.Gorodilov, V.Morozenko Genetic Algorithm for Finding the Key's Length and Cryptanalysis the Permutation Cipher. International Journal "Information Theories & Applications. Vol.15/2008.
7. D. Kahn The Codebreakers. Scribner 1996.
8. J. H. Holland Adaption in Natural and Artificial Systems. Univ. Of Michigan Press. 1975.
9. Goldberg D.E. Cenetic Algorithms in Search, Optimization and Machine Learning. Univ. of Michigan Press, 1989.
10. S. R. Baraga, P. S. Reddy A Survey Cryptanalytic Works Based on Genetic Algorithms. IJETTCS, Vol.2. Issue 5, September –October, 2013.

Article received: 2017-09-28