Cipher Cracking with Genetic Algorithms

Mayr Michael, BSc.
University of Applied Sciences Upper Austria
Data Science
Softwarepark 11
4232 Hagenberg, Austria
s1710595012@students.fh-hagenberg.at

ABSTRACT

This paper describes the performance of modern genetic algorithms concerning the decryption of monoalphabetic (Caesar) and polyalphabetic (Vigenere) substitution ciphers. Appropriate mutation and crossover operators as well as different approaches of evaluating a solution candidate are discussed.

Keywords

Genetic Algorithms, Substitution Cipher, Crypto Analysis, Caesar Cipher, Vigenere Cipher

General Terms

Algorithms, Measurement, Experimentation, Performance

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Heuristic methods; G.1.6 [**Numerical Analysis**]: Optimization; E.3 [**Data**]: Data Encryption

1. INTRODUCTION

The objective of this paper is to examine the possible applications of genetic algorithms in cryptology. Ciphers are the main component for encrypting messages. Since the early stages of cryptography development different ciphers have become used in practice. The complexity and security of ciphers differ greatly. Substitution and transposition ciphers are the simplest forms of encryption. Substitution cipher replaces symbols of a plaintext with ciphertext, according to a fixed system. A well known technique used hundreds of years ago is the Caesar cipher which counts as a simple monoalphabetic substitution cipher. A polyalphabetic approach, by contrast, uses more than one monoalphabetic cipher within the same text. A well known example is the Vigenere Cipher. There are several traditional mathematical programming implementations regarding the decryption of the above mentioned substitution ciphers however those tend to lead to high computational costs. If the GA-based approach proves successful, it could lead to faster, more automated cryptanalysis. However traditional techniques will likely score better results than the GA-based approach because the latter is hardly researched in the field of cryptanalysis. If the GA-based approach proves unsuccessful on the simple substitution ciphers, it is not worthwhile to apply the same approach to more modern and complex ciphers like DES or RSA.

2. METHODS

2.1 Cryptology

This section describes two simple substitution cipher which are used in this paper.

2.1.1 *Monoalphbetic Substitution Cipher*

A monoalphabetic substitution cipher is one where $F_{\mathrm{k}}(m)$ is a simple substitution function which replaces every $m \in M$ with a corresponding $c \in C$ according to the cipher key k [3]. The most obvious substitution cipher is the Caesar cipher, which was in fact used by Julius Caesar to for communication. Caesar shifted shifted the alphabet by three which is $k = C$. That means that Caesar cipher defines $F_{\mathrm{k}}(m)$ as the mapping seen in the following.

$$\begin{bmatrix} b \in m & : & \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} & \mathrm{E} & \mathrm{F} & \mathrm{G} & \mathrm{H} & \mathrm{I} & \mathrm{J} & \mathrm{K} & \mathrm{L} & \mathrm{M} \\ & & \mathrm{N} & \mathrm{O} & \mathrm{P} & \mathrm{Q} & \mathrm{R} & \mathrm{S} & \mathrm{T} & \mathrm{U} & \mathrm{V} & \mathrm{W} & \mathrm{X} & \mathrm{Y} & \mathrm{Z} \\ \mathcal{E}_k(b) & : & \mathrm{D} & \mathrm{E} & \mathrm{F} & \mathrm{G} & \mathrm{H} & \mathrm{I} & \mathrm{J} & \mathrm{K} & \mathrm{L} & \mathrm{M} & \mathrm{N} & \mathrm{O} & \mathrm{P} \\ & & \mathrm{Q} & \mathrm{R} & \mathrm{S} & \mathrm{T} & \mathrm{U} & \mathrm{V} & \mathrm{W} & \mathrm{X} & \mathrm{Y} & \mathrm{Z} & \mathrm{A} & \mathrm{B} & \mathrm{C} \end{bmatrix}.$$

Figure 1: Caesar cipher

2.1.2 *Polyalphebtic Substitution Cipher*

A polyalphabetic substitution cipher is simply multiple monoalphabetic substitution ciphers applied to the same plaintext message m. By far, the best known polyalphabetic substitution ciphers is the Vigenere cipher [3].

2.2 Genetic Algorithm

The genetic algorithm does not always produce the exact answer, but rather it gives a solution that is close to the correct one. In the case of deciphering the ciphertext, after using the best key produced by genetic algorithm, it is probably easy for a human to read the decrypted text. Each solution stores a key k of length n. The mutation and crossover operators alter k.

Computational Intelligence 2, 2018, FH O, Hagenberg.

2.2.1 Operators

Operators select, mutate or crossover solution candidates in different ways. It is not exactly clear which operators provide the best results. Some possible methods are discussed in the following. Many operators also rely on access to language statistics for quality improvement.

Selection.

In the selection step 2 good solutions are selected from a population of solutions. Two ways of selecting good solutions are the following and are tested in this paper:

- Best: Selects the best solution.
- Fitnessproportional: Every solutions is given a chance of being selected to crossover but fitter candidates are more likely to be chosen than weaker individuals.
- Tournament: The selection happens in tournament form. The winner of each tournament is selected for crossover.

Mutation.

Two mutation operators are used to cryptanalyze the ciphers. A combination as well as each seperate mutation will be tested.

- Random character: Chooses a position in the key and replaces it with a random character
- Shift character: Chooses a position in the key and replaces it with the next variable in the alphabet.

Crossover.

The crossover operator merges two solutions and produces two child solutions. The following crossover operators are tested:

- Single Point: Chooses a random position and swaps the two characters.
- Double Point: Chooses two random positions and swaps the interval of characters.
- Double Point Ordered: Chooses two random positions and swaps the interval of characters. Same characters are not allowed in the crossover.

2.2.2 Fitnessfunction

A big problem with using genetic algorithms for decryption is the fitness function. It is not trivial to implement a good fitness function.*Language statistics* and *dictionary attacks* should counter this problem. In this case the fitness function evaluates key of length n by decrypting the ciphertext with the key and calculating the statistical differences of plaintext and decrypted text. A very interesting approach for calculating the fitness is described in a paper for genetic optimizations from Andrew Clark. [1]. This approach will be tested and if successful it will be paired with the dictionary attack.

2.3 Attacks

This section describes different approaches attacking the fitness function problem.

2.3.1 Language statistic

The statistic is a measure of the redundancy found in a piece of text. English, like other languages has its own characteristic redundancy [2]. All examples in this paper are based on English plaintext. This method relies on the fact that the plaintext language must be known beforehand.

2.3.2 Dictionaryattack

The goal of using a dictionary is to improve the decryption rate of the genetic algorithm. Attacks solely based on language statistics might not prove successful. The dictionary stores common English words and can be used to scan for known words in the (partly) decrypted ciphertext.

3. CONCLUSION

Currently this paper is still in work, a final conclusion will be available in a few weeks. Despite that, it is expected that the genetic algorithm performs good and fast on simple ciphers like Caesar. It is very likely that the algorithm fails on more complex ciphers with higher key length. I am confident that genetic algorithms can crack polyalphabetic substitution ciphers like the Vigenere up to a key length five.

4. REFERENCES

[1] A. Clark. *Modern optimisation algorithms for cryptanalysis.* Nov 1994.

[2] W. F. Friedman. *The Index of Coincidence and Its Applications in Cryptography.* L. Fournier, 1922.

[3] W. Mao. *Modern Cryptography - Theory and Practice.* Prentice Hall, London, 2003.