Breaking a polyalphabetic substitution cipher using coincidence index and frequency analysis

Gabriel Oliveira
Computer Science School
Pontifcia Universidade Catolica
do Rio Grande do Sul Porto Alegre, RS, Brasil
Email: gabriel.pimentel@acad.pucrs.br

***Abstract*—Frequency analysis is a method to study of the frequency of occurrence of letters in a text. The method is used as an aid to breaking classical monoalphabetic ciphers, but it is not useful against polyalphabetic ciphers. The index of coincidence is the technique used to measure of how likely it is to draw two matching letters by randomly selecting two letters from a given text. In this text, we will show how we had used those techniques to break a polyalphabetic substitution cypher.**

Keywords*—*Cryptography, Frequency Analysis, Index of Coincidence.

I. Introduction

The field of cryptography is not only filled with monoalphabetic ciphers, the ones we move all the letters by a certain amount of positions. One attacker can mix many text movements using a proper key and the Vigenre cipher [5]. This paper described a technique to break those kind of ciphers.

A. Frequency Analysis

Many techniques use statistical properties of the English language [2]. Frequency Analysis is one of those techniques, which relies on the count of occurrences of letters in the text or the frequency of their appearance to extract patterns from the text. It‘s often used to decipher texts that were encoded using a substitution cipher (explained in [2]) - also known as monoalphabetic ciphers.

In this work, we had used the English letters frequency obtained from [1] and [3].

B. Index of Coincidence

Coincidence counting is the technique [4] of putting two texts side-by-side and counting the number of times that identical letters appear in the same position in both texts. This count, either as a ratio of the total or normalized by dividing by the expected count for a random source model, is known as the index of coincidence (IC).

To obtain the IC of a given text, we use the following formula, where N is the length of the text and n_1 through n_c are the frequencies (as integers) of the c letters of the alphabet (c = 26 for monocase English). More details on how this formula was obtained can be found on [4].

$$IC = \frac{\sum_{i=1}^{c} n*(n-1)}{N*(N-1)}$$

II. Combining Frequency Analysis and the Index of Coincidence

Since the frequency analysis technique is only used on monoalphabetic ciphers, our proposal is to combine it with the index of coincidence technique in order to decipher texts encrypted with polyalphabetic ciphers.

The Vigenre cipher [5] is one example of polyalphabetic ciphers. It uses a key to encrypt the text and the same one to decrypt it. Without any knowledge about the key used to encrypt the text it is not possible to read it.The index of coincidence can help us to start solving the Vigenre cipher by helping us to discover the length of the key.

A. Finding the Key Lenght by dividing the text in columns

One way to do that is to guess the key length and use the index of coincidence to validate our guess. Intuitively, we know that the key is probably smaller than the text. Thus, a single character of the key is used to encrypt many characters in the text. If we guessed the key length correctly, all those characters encrypted with the same character of the key constitute a part of the text on witch we should see the same IC of an English text. Also, all those characters encrypted with the same character of the key constitute a monoalphabetic ciphered text and can be deciphered using the frequency analysis technique. An algorithm to follow those steps can be found below:

```
function find_best_key_len(m):
    english_ic = 0.067
    ics = {}
    foreach key_lenght k do:
        columns = break_message(m, k)
        column_ic = {}
        foreach columns c do:
            column_ic += calculate_ic(c)
        end
        key_ic = avg(column_ic)
        ics += (key_ic - english_ic)
    end
    return min(ics)
```

Note that when using such algorithm, the described formula for the index of coincidence should use N as the column number of characters. Also note how the value 0.067 (taken from [2]) is being used as reference to the index of coincidence of a normal english text.

B. *Finding the Key Length using the whole text*

Another way to do that is to apply the Friedman test in all text. This is a variant way to estimate a cipher key lenght and can be calculated as follows:

$$h \approx \frac{(E_s - \frac{1}{N}) * k}{(k-1) * \phi(T) - k * \frac{1}{N} + E_s}$$

Where E_s is the expected coincidence index for our natural language (0.067 as already mentioned), $\phi(T)$ the total coincidence index of our text, N the alphabet size, k the text length.

C. *Frequency analysis on a polyalphabetic text with a key of size k*

After the key lenght was found (by either the methods described before), one can apply frequency analysis on each column to decode the text using the following algorithm

```
function move_letters_with_key_len(m,kl):
    columns = break_message(m, kl)
    plain_columns = {}
    key_decrypted = {}
    foreach columns c do:
        k,t = move_letters(c)
        plain_columns += t
        key_decrypted += k
    end
    text = assemble_columns(plain_columns)
    key = key_decrypted.join()
    return (text,key)
```

III. Conclusion

This paper have showed how to proper deal with a polyalphabetic cipher, such as the Vigenre cipher [5], using the index of coincidence and the Friedman test technique.

The cipher text used to validate those algorithms were using the 5-letter-key *PLATO* and started with the following phrase (spaces and capital letters were used here to facilitate the reading): *Neither must we forget that the republic is but the third part of a still larger design which was to have included an ideal history of Athens as well as a political and physical philosophy*

References

[1] http://en.algoritmy.net/article/40379/Letter-frequency-English, accessed in March 27th 2016

[2] Douglas R. Stinson, *Cryptography: Theory and Practice, Third Edition*, 2005.

[3] Robert. Lewand, *Cryptological mathematics*, Mathematical Association of America Textbooks, 2000.

[4] Friedman, W.F., *The index of coincidence and its applications in cryptology*, Department of Ciphers. Publ 22. Geneva, Illinois, USA: Riverbank Laboratories. OCLC 55786052. The original application ignored normalization, 1922.

[5] Bruen, Aiden A., Mario A., *Cryptography, Information Theory, and Error-Correction: A Handbook to the 21st Century*, John Wiley and Sons, ISBN 978-1-118-03138-4, 2011.

[6] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.