Security
Frequential analysis

R. Absil - M. Wahid

Academic year 2019 - 2020

The goal of this homework is to make you implement the cryptanalysis of the Cesar and Vigenère ciphers, with frequential analysis. You have to submit your work on 29 March at 23h55.

1 Objectives

You must, from a sufficiently large text file, implement an *automatic* cryptanalysis of the Cesar and Vigenère ciphers. For that purpose, you must use a frequential analysis attack, as described in the course. Consequently, you code must allow to

- encode and decode text (plain or not) using these ciphers with the help of a key,
- decode ciphered text (ciphered with any of these algorithms) *without* the key.

Not that in both cases, it is possible that the "e" is not the most frequent letter in the plain text. Consequently, it is probably useful that your application consider this case automatically[1].

Note that considering this howework is very short and basically consists in coding scripts rather and implementing a "true" project, you don't have to document your code more that what the minimum requires, or submit a report.

On the other side, the choice of programming language is left to your discretion (you take responsibility for this choice).

[1] For that purpose, you will probably have to implement concepts relative to statistical analysis, namely the coincidence index, and the χ^2 test ("Chi 2 test"). Note that you don't have to *master* these concepts: just kow when to use them and why, and correctly implement them.

🅭🅯🄏🄎

2 Text and preprocessing

You can find large texts on Gutenberg.org. Note that, for simplicity reasons, you are allowed to preprocess your plain texts, for instance to delete spaces[2] and diacritics.

3 Submission

The minimal requirements for submitted projecst are as follows:

- projects have to be submitted on time on PoÉsi, that is, maximum on 29 March at 23h55,
- projects have to be submitted under `.7z`, `.zip`, `.tar` or `.tar.gz` format,
- projects have to provide a readme file explaining
  - how to use your project (for example, to run the cryptnalysis of Vigenere, type the following command in a shell)
  - how to build your project (we recommand here to either provide a makefile, or a shell script to install missing dependencies, compile the project and run relevent sripts).

Projects failing to meet these requirements will not be grade (that is, they will get 0/20).

[2] Whatever space it is, such as regular space, tabulation, etc.