M1 MoSIG : Crypto TP1 Blaise Vigenere's system.

SID-LAKHDAR Riyane

2012 - 2013

Abstract

For this lab, we have implemented 4 java programs within the same package. To be executed, this programs need to be lunched next to the provided directory "resource" as follows:

1. java VigenereCipher <name of the clear text> <name of the output ciphered text> <key>
2. java VigenereDecipher <name of ciphered text> <name of the output clear text>
3. java VigenereKeyGuess <name of ciphered text>

Contents

1 Vigenere code 2

1.1 Numerize text 2

1.2 Cipher 2

1.3 Decipher 2

2 Key size 2

2.1 Occurrence of letters 2

2.2 Coincidence index 2

2.3 Friedman's method 2

2.4 Results 3

1 Vigenere code

1.1 Numerize text

1. File "src/crypto/CryptoTools" function "numerizeText"

1.2 Cipher

1. File "src/crypto/Vigenere" function "cipher"
2. File "src/crypto/VigenereCipher" function "main"

1.3 Decipher

1. File "src/crypto/Vigenere" function "cipher"
2. File "src/crypto/VigenereDecipher" function "main"

2 Key size

2.1 Occurrence of letters

1. File "src/crypto/CryptoTools" function "getNbrOccurence"

2.2 Coincidence index

The coincidence index provides a measure of how likely it is to draw two matching letters by randomly selecting two letters from a given text (cf Wikipedia). Thus, for a given text, this index Kapa is defined as the some for all the possible characters of the probability to draw two time this character:

$$\kappa = \sum_{c} \frac{occurrence(c)}{n} * \frac{occurrence(c) - 1}{n - 1} \tag{1}$$

2.3 Friedman's method

Let l the size of the expected key. And let c_1 and c_2 two random characters in the ciphered text at the indexes i_1 and i_2.
Using the definition of the coincidence index κ, we have:

$$\begin{aligned} \kappa &= Proba(c_1 \text{ and } c_2 \text{ are the same and } i_1 = i_2[l]) \\ &+ Proba(c_1 \text{ and } c_2 \text{ are the same and } i_1 \neq i_2[l]) \\ \kappa &= (\frac{\frac{n}{l} - 1}{n - 1}) * P_m + (1 - \frac{\frac{n}{l} - 1}{n - 1}) * P_p \end{aligned}$$

Thus:

$$l = \frac{n * (P_p - P_m)}{kapa * (n - 1) + Pp - Pm * n}$$

Where P_m is the probability to draw 2 same characters in an non equi-distributed alphabet (Mono alphabetic cipher) and P_p in an equi-distributed alphabet (Poly alphabetic cipher). We can deduce the estimated size of the key $l = \frac{n*(P_p-P_m)}{kapa*(n-1)+Pp-Pm*n}$

The corresponding program has be implemented in the file "src/crypto/Vigenere.java" function "keySizeApproximation".

2.4 Results

Running our program on the text "acrypter.txt" returns a key of size 0. Which is coherent with the fact that this text has not been ciphered