Cryptology – Lab 1

Cryptoanalysis of the Vigenere cipher

Frédéric Haziza <daz@it.uu.se>

Department of Computer Systems

Uppsala University

Spring 2008

Deliveries

- Work in group of 2
- Oral examination
  - Show your code and tests
  - Describe your implementation
  - Demonstrate your *working* version
- Date: February 7th and 8th
- Deadline: February 13th at 17.00
- Examination: February 14th and 15th at 10.00

Goal

Find plaintexts and keys for the 6 given cryptotexts

Help files and description in [HOMEPAGE$_{cryptology}$]/labs

Lab 1 – Vigenère Cipher

- Encrypts m characters at a time
- Key word $\mathbf{K} = (k_1, k_2, \ldots, k_m)$
- Encryption:
  $E_K(x_1, x_2, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$
- Decryption:
  $D_K(x_1, x_2, \ldots, x_m) = (x_1 - k_1, x_2 - k_2, \ldots, x_m - k_m)$
- Operations modulo 26

Example

Plaintext: “Renaissance”
Key word: “Band”

Plaintext	RENA ISSA NCE
Key	BAND BAND BAN
Ciphertext	SEAD JSFD OCR

Cryptoanalysis

Step 1

Determines the key length m

Step 2

Determines the key $\mathbf{K} = (k_1, k_2, \ldots, k_m)$

Determining the key length

Method 1: *Kasiski test*

1863 - Major F.W. Kasiski, German cryptologist

Length of keyword is a divisor of the gcd of the distances between identical strings of length at least 3.

Kasiski test

Observation

2 identical segments of plaintext will be encrypted to the same ciphertext whenever their occurence in the plaintext in δ positions apart, where $\delta \equiv 0[m]$

- Search the cipher for pairs of identical segments of length at least three
- Record the distance between the starting positions of the 2 segments
- if we obtain then $\delta_1, \delta_2, \ldots$
- then we conjuncture that m divides each δ_i and hence m divides their gcd.

Determining the key length

Method 2: *Sliding stripes*

- Write the entire cryptogram on two separate strips of paper.
- Line up the strips, one above the other and slide the top strip one position at a time.
- At each step count the number of agreements of letters in the two strips.
- The slide position that gives the greatest number of agreements is usually the key length

Determining the key length

Method 3: *Kappa test*

Friedman Test (1925, Colonel William Frederick Friedman (1891-1969))

Uses the *index of coincidence*

We'll use that one

Index of Coincidence

index of coincidence of a text $I_c(\mathrm{X})$

probability that a randomly chosen pair of letters in the message are equal.

- Text: $\mathbf{X} = (x_1, x_2, \ldots, x_n)$
- Text length: n
- n_i number of occurrences of the ith letter in $\mathbf{X}$

$$I_c(\mathrm{X}) = \sum_{i=0}^{25} \frac{n_i(n_i-1)}{n(n-1)}$$

Index of Coincidence

We can also calculate this index for any language source:

- p_a = probability of occurrence of the letter a

  its frequency in the alphabet

- $I_{source} = p_a p_a + p_b p_b + \ldots + p_z p_z = \sum_{i=0}^{25} p_i^2$

$I_{English} \sim 0.065$

$I_{Random} = \sum_{i=0}^{25} p_i^2 = \sum_{i=0}^{25} \left(\frac{1}{26}\right)^2 = 26 \left(\frac{1}{26}\right)^2 = \frac{1}{26} \sim 0.038$

Using the Index of Coincidence

This index can give information about a message.

For instance, if a ciphered message was either a transposition or a monoalphabetic substitution then one would expect to have $I_{Message} \sim I_{English}$, but if a polyalphabetic substitution was used then this value should decrease (but no lower than 0.038) since the polyalphabetic procedure tends to randomize the occurrences of the letters.

Rearrange the text

$$
\begin{aligned}
y_1 &= x_1 x_{m+1} x_{2m+1} \cdots, \\
y_2 &= x_2 x_{m+2} x_{2m+2} \cdots, \\
\vdots &\quad \vdots \quad \vdots \\
y_m &= x_m x_{2m} x_{3m} \cdots
\end{aligned}
$$

If m was indeed the key length, $I_c(y_i) \sim 0.065$

However,

$$m = \frac{0.027 * n}{(n-1) * I_c(\mathrm{X}) - 0.038 * n + 0.065}$$

Assume we know the key length

To find key $\mathbf{K} = (k_1, k_2, \ldots, k_m)$

- For each y_i and for each letter α from A to Z
  - Decrypt y_i with α using the Ceasar method,
  - Measure frequencies of letters in the decrypted substring
  - Compare it to the english distribution
    (define your distance, for eg, sum of squares)
- For the minimal deviation, $k_i = \alpha$
- Repeat for each k_i

When we get the key

- Decrypt the cipher **X** using the key **K**,
- Check the $I_C(\mathrm{X})$
- If it differs too much from $I_{English} \sim 0.065$
  - try the suggested key length ± 1
  - Repeat until we get close to 0.065
- Suggested range: 0.056 – 0.075