Coursework for CSC3621 Cryptography

Part I (30/100)

This is the first of the three parts of the coursework for CSC 3621 Cryptography. It consists of the following three exercises with each exercise 10 marks.

Submission instruction

- Submit all three exercises together in one compressed file (.zip), which should contain three clearly separated directories (say Exercise1, Exercise2 and Exercise3) with each directory containing the report and relevant source code files (.java).
- Deadline: **Friday, 17:00, 30 Oct, 2015**

Exercise 1 (10 marks):

Aim: To understand the non-uniform distribution of English letters and how that may be exploited in cryptanalysis.

Instruction: Write a Java program to run a frequency analysis of English text. The input to the program should be a relatively long English text file. You can take the following file as the input:

http://homepages.cs.ncl.ac.uk/feng.hao/teaching/pg1661.txt

The program counts the occurrences of the 26 English letters from A to Z (case insensitive). You are also encouraged to try different English files, and see if you get similar results.

Based on the above frequency analysis, cryptanalyze the following ciphertext and recover the plaintext:

http://homepages.cs.ncl.ac.uk/feng.hao/teaching/Exercise1Ciphertext.txt

Submission guideline: Write a report (not more than 3 pages) about the findings from your program with screenshots. Attach the source code as separate Java files in the submission. Compare the letter frequencies you obtained with the commonly known results (http://en.wikipedia.org/wiki/Letter_frequency). Comment on the difference if any. Explain the steps you did to cryptanalyze the given ciphertext.

Marking guideline:

- A working frequency analysis program with correct result – 2 marks
- A working cryptanalysis program with correct result – 2 marks
- A concise and insightful report – 6 mark

Exercise 2 (10 marks):

Aim: To understand the working of a polyalphabetic cipher and its weakness. In particular, the Vigenère cipher will be studied.

Instruction: Write a Java program that accepts a user-defined password (letters only) as the key and encrypts the following text file using a Vigenère cipher.

http://homepages.cs.ncl.ac.uk/feng.hao/teaching/pg1661.txt

With the same key, the program should be able to decrypt the ciphertext. Obtain the letter frequencies of the ciphertext and compare those with the results obtained in last week's exercise. (Tip: for simplicity, treat all English letters as lower-case. For characters that are no letters, just keep them intact during encryption and decryption.)

To test the correctness of your program, try to use "ncl" as the key to encrypt a file that contains a string "newcastleuniversity". Compare your results with your classmates.

Given the knowledge that the following ciphertext file was generated by using a Vigenère cipher, try to recover both the plaintext and the encryption key. (Tip: non-English letters such as spaces, punctuations etc have been removed from the plaintext file before encryption.)

http://homepages.cs.ncl.ac.uk/feng.hao/teaching/Exercise2Ciphertext.txt

Submission guideline: Write a report (not more than 4 pages) about the programs you have written with screenshots; attach the source code in the submission. Discuss to what extent has the Vigenère encryption changed the distribution of letter frequencies. Explain the steps you did to cryptanalyze the given ciphertext. Note that the program is just a tool to help cryptanalysis and verification; it's the reasoning process in breaking the cipher that matters in the report.

Marking:

- A working Vingere encryption program – 3 marks
- A working Vingere cryptanalysis program (manual tuning is allowed; not need to be fully automated) – 3 marks
- A concise and insightful report – 4 mark

Exercise 3 (10 marks):

Aim: To understand the working of a one-time pad and practice a two-time pad attack.

Instruction: Write a program (preferably Java) to generate a one-time pad, which is a relatively large file of all random data (say 1 MB). The program should also be able to encrypt/decrypt files based on the generated one time pad. Work in pairs. Find a partner in the class and exchange with him/her a one-time pad, so both have the same shared secret. Send the encrypted file to your partner (using USB or email), and see if your partner can decrypt to the same result.

Tip: use the following test vector to check if your program does encryption correctly.

```
Plaintext (ASCII):  Every cloud has a silver lining
OTP (HEX):          6dc72fc595e35dcd38c05dca2a0d2dbd8e2df20b129b2cfa29ad17972922a2
ciphertext (HEX):   28b14ab7ecc33ea157b539ea426c5e9def0d81627eed498809c17ef9404cc5
```

The following ciphertexts were generated by re-using the same one time pad. Your goal is to decrypt the last ciphertext.

```
Ciphertext1 (HEX):  dcb68a9df8f716409ba0fb55ee3fc8b91f090177976e0961
Ciphertext2 (HEX):  d4e2c992e9a11e53c2f2f653ef27c8ba1e5d403e882717699d852c
Ciphertext3 (HEX):  d1ff8299acb11a558ae5e51aed3d83bd4b5a0f39
Ciphertext4 (HEX):  d0f78785acb65b4d8bf4e356e47485b9004c13779d270a6f8b9c3314
Ciphertext5 (HEX):  c9fe8cdcf8a50e558aa0e053ed38c8b71e5d
Ciphertext6 (HEX):  c9f9c999fab2095896e8fe54e6749cb00e5b057795744767c8853a1069c6be
Ciphertext7 (HEX):  c4f99c88e4f71252c2f7f649f5318cf8044740239462477f87823116
Target ciphertext:  c4f99cdce4b60d44c2e4f854e47491b71e5b402093750c
```

Tips:

- *Consider what happens when a space is XORed with a character in [a-zA-Z].*
- *Refer to a sample attack program: http://homepages.cs.ncl.ac.uk/feng.hao/teaching/OTPAttack.java. The output of the program should give you sufficient information needed to crack the target ciphertext.*

Submission guideline: Write a report (not more than 5 pages) about the one-time pad program you have written with screenshots of the output. Attach the source code in the submission. State if you and your partner can decrypt each other's ciphertext correctly. Explain in the report the steps you took to cryptanalyze the ciphertexts that were generated from re-using the same one-time pad key and how you have cracked the target ciphertext.

Marking:

- One-time pad generation/encryption/decryption program – 4 marks
- A concise report that explains clearly how the above program works and how you crack the target ciphertext – 6 marks