CS 478/513: Computer Security
Spring 2022
Programming Assignment 1
Due: Tue., 3/2/22, 11:59 pm

Overview All code must be written modularly in C or C++, and must successfully compile and run on the Linux machines in the CS department's labs. This problem is equivalent to Problems 11 - 13 from Chapter 2 of the textbook. However, you need not follow the exact algorithm proposed for Problem 13. You may improve it if you choose to do so.

The provided file `ciphertext.txt` contains a message which has been encrypted using a simple substitution cipher. You are to create a program which can assist a cryptanalyst by providing the following functionality:

1. Compute and display the frequency analysis of the ciphertext.
2. Decrypt the ciphertext using a key provided by the analyst.
3. Guess the key of the cipher.

In order to guess the key, the following procedure should be followed:

1. Compute the frequency ordering of the letters appearing in the ciphertext, and guess a potential decryption key by mapping them to the standard English letter frequency ordering: **ETAOINSHRDLCUMWFGYPBVKJXQZ**.
2. Analyze the quality of your potential key by counting the number of words from the provided `dictionary.txt` file which appear in the decoded message.
3. Permute the guessed key in order to increase the number of dictionary words the decrypted text contains, until further permutations no longer improve its quality.

Your program should print out the final key that it computed as well as the resulting decoded message. Do not forget that your program must also be able to facilitate manual cryptanalysis as described above – your program is not expected to reproduce the exact plaintext, but rather get close enough such that an analyst (or you, in this case) can easily determine the true key.

Hints and Suggestions:

- The textbook suggests shifting letters from the frequency distribution by one position in order to improve the guessed key. You can also try shifting letters by more than one position if it yields a better result. In fact, it may possible to decode the given ciphertext completely, without human intervention, by applying this suggestion.
- Instead of scoring your key by only the number of dictionary words it produces in the deciphered text, you can also try to factor the length of detected words into the score.

Submission requirements: Upload a tarball (.tar or .tar.gz) to Canvas. The submission should include the following material:

1. Your C/C++ code, with a Makefile to compile it.
2. A text file (Readme) explaining how to use your program for each of the following tasks:
   (a) Display the frequency analysis of the ciphertext.
   (b) Use the program to guess the cipher key and decrypt the text using it.
   (c) Decrypt the text using a key entered by the user.

   Your Readme should also explain the algorithm you used to guess the key, if it is different from the one proposed in the textbook.

3. A text file containing the decryption key you obtained and the plaintext it produced. If manual cryptanalysis was required after running your program, you should also explain the steps you took to finish decrypting the message.