ECE-458 (Spring 2023) – Assignment 1

Due on 2023-06-02, 23:00 (Friday, 11:00pm)

Instructions

Your solutions should be submitted through LEARN by the specified due date, and they must be uploaded as a *plain text file*. For example, if you work on it on Windows, you could use Notepad or some text editor that you use for programming. If on Linux, you could use gedit or kwrite or Eclipse or KDevelop (any text editor).

You are allowed to work with your classmates and discuss ideas and solutions; however, the submitted work MUST BE STRICTLY YOUR OWN WRITING. This also applies to looking up ideas on the Internet or textbooks. Moreover, any collaboration MUST be acknowledged in the submission (explicitly name the persons that helped you or that you helped, persons that shared ideas with you or that you shared ideas with them; explicitly name online sources or textbooks where you found ideas that helped you write your solution, etc.).

The standard advice on what to do about collaborations to avoid getting in trouble is: you may discuss and share ideas with classmates, but then, put aside any annotations and DO NOT LOOK at them when writing your own solution (this ensures that your writing is original and distinct). Also, NEVER share any of your written solutions or your written code with anyone; preferably, DO NOT ALLOW ANYONE to look at your written solutions (keep in mind that if they have good memory and write the exact text in your solutions and submit it, you will be equally liable in the academic dishonesty incident — see Course Outline).

It is assumed that you have carefully read the additional conditions included in the Course Outline (available through LEARN) regarding academic integrity, assignments submissions, etc.

Breaking the Vigenère Cipher

In this assignment, you will design and apply a technique to cryptanalyze the Vigenère Cipher. You will be given a certain amount of ciphertext (approx. 15 kilobytes of ciphertext) that has been encrypted with a key of unknown length.

The plaintext will be subject to a random permutation of the lines — a different permutation for each student. This way, the statistics of the text will be the same for everyone, and we'll have the advantage that LEARN will not think that the submission is plagiarized (as would happen if all students submit the exact same plaintext that they got after breaking the cipher).

To download your ciphertext, just follow the link to "Assignment 1 ciphertext" posted on LEARN under the Assignment 1 module. In that page, you enter your UW username and your student ID. Download *exclusively* your own ciphertext.

The convention for the text (both in plaintext and ciphertext) will be the same as described in the lecture slides (no spaces or punctuation, no numeric digits; only letters from the English alphabet in lowercase, etc.)

Please notice: the ciphertext file does not have spaces, newlines, etc. It will be a text file with a single and *very* long line. Your text editor may or may not be able to display the file. Technically, you do not even need to look at the file; but if you want to, on Linux you could use the command `more yourfile.txt` and it will pause as soon as the characters fill one screen.

Specific Requirements / Deliverables:

- You are not allowed to use the index of coincidence to determine the key length; instead, you must display the statistics (the letter frequencies) every N positions, and try different values of N until the displayed statistics show that you have the correct key length.
- You must write and submit programs to:
    - Perform encryption and decryption function (you may notice that one function may suffice, if you aply a simple conversion to the key).
    - Display the statistics (the letter frequencies) of the ciphertext for subsequences of one every N characters.

For the programs, using C++ would be preferred, but you are allowed to use either C, C++, Java, or Python. No other programming languages will be allowed. Please notice, **you must write your own programs**, and are not allowed to use any code found on textbooks or online.

You must also submit:

- A report specifying the steps that you followed to obtain the key length, and then to obtain each of the letters of the key. You should include the "evidence" for your reasoning and conclusions. For example, if you guessed that one particular letter of the key is **b** because the letter frequencies showed that the letter **f** appears, say, 13.1% of the time in the ciphertext, then you should show the output of your program showing the frequency of the letters.
- The plaintext (as a text file) — you can either write your program such that the decrypted text is output to a file, or just write to stdout and redirect to a file once you completed the cryptanalysis.

The grading scheme is: 50% for the submitted programs; 25% for the cryptanalysis strategies related to finding the key length; 25% for the cryptanalysis strategies related to finding each of the letters of the key.