Chad Reynolds
CS:4640 Computer Security
Homework 1
Due: February 22, 2016

Overview

Goal:
Implement two encryption algorithms out of the three possible choices:

- Vigenère cipher*
- Permutation cipher*
- Simple Enigma machine with one rotor

* = my choice

The assignment requires the encryption to be performed on two different plaintext, in parts 1 and 2. In part 1 a pre-processing step requires that all non-alphabetic or space characters be removed, and that the alphabetic characters be converted to lowercase. In part 2 this pre-processing step is extended to also remove spaces.

See Assignment_1.pdf for more specific details.

Algorithm Descriptions

Vigenère

The Vigenère cipher works in a similar fashion to the classic Caesar cipher. The Caesar cipher takes a character from an alphabet, then uses its index within that alphabet as a shift amount for each character in the plaintext. Using the English alphabet as an example with the plaintext "hello", key of b (a shift of 1), then the ciphertext becomes "ifmmp".

The Vigenère cipher extends this idea by accepting a key longer than one character, providing a different shift amount for each character in the key. If the key is shorter than the plaintext, then the key is repeated a number of times until the entire text can be shifted.

Permutation

The Permutation cipher relies solely on transposition and not substitution like the Vigenère. Letters are shuffled within blocks according to the key. A key σ maps characters original positions to their new positions. Again using the plaintext "hello", with the key of $\sigma = (\ 1\ 0\ 2\ 3\ 4\ )$, our plaintext becomes "ehllo".

We can see that all of the same letters are in place, but the character in index 0 has moved to index 1, and vice-versa. The indexes can be shuffled around in any order in the key, keeping in mind that certain keys (such as this one) will not be particularly effective at obscuring the meaning of the plaintext.

Installation and Running instructions

Installation

The "installation" is as simple as downloading the files to a location on your computer. However, the algorithms are coded in the Python programming language and thus depend on a Python installation being present. Specifically, I tested this with Python version 3.5.0 but I believe any Python3 installation should be capable of running the code.

Find up-to-date install instructions for most platforms on the Python website: https://www.python.org/downloads/

Running the tool

This is a command-line tool, and assumes a bash shell. I believe the Python argument parsing library should run without issue in a Windows command-prompt but have not tested it. There may be an issue with options.

To run the tool, assuming python3 is on your PATH, and the current directory is the project directory:
python3 src/main.py input_file key_file vigenere | permutation
(where vigenere | permutation is the choice of algorithm)

This will cause the algorithm output to be printed to standard out. There are options/flags for: useful help text(-h, –help), switching to decryption(-d, –decrypt), and switching to part 2's no-space pre-processing(-n, –no-spaces)