Carl Bunt
Email: cbunt@psu.edu
Course: CS 491 - Intro to Security
Instructor: Dr. Karen L. Karavanic

Homework - 2
ID: 915910838
Term: Fall 2020
Due Date: 18 OCT 20

Question A:
Written Exercises

1. What is the difference between private and public key cryptography? For each, give one scenario where we might use it today.
2. Given a key with 64 bits:
   a. how many trials of exhaustive search must be done before success is expected?
   b. Assume each exhaustive search trial requires a total of 16 floating point operations. Is a 64 bit key length a reasonable choice given today's most powerful supercomputer?

Response:

1. Private Key vs. Public Key Cryptography

Private Key is a symmetric encryption method. This means the same key is used for both encryption and decryption. Symmetric encrtption is typocally used for encrypting local data storage such as a zip file where the same user will be accessing and manipulating the encypted data.

Public Key is an asymmetric encryption method. This means there are to encryption keys. One used encryption and a differnt for decryption. Asymmetric encryption is typocally used for encrypting communications where unone user will be generating and encypting the data and a differnt user will be decrupting and reading the data.

2. Given a key with 64 bits:
   a. A 64-bit key would 2^{64} brute-fource attempts before an exhaustive key search can be completed.
   b. The weakest super computer listed on the TOP500 list is rated at $1225 TFlop/s$ so even the weakest computer could exahstivly trial all possible keys in 2.7 days making 64-keys an unwise length for any form of secure encryption.

Question B:
Hands On: Vigenère Ciphers
As a "warmup" to C programming, we will have some fun and implement a simple form of encryption called Vigenère Ciphers. In addition to a C refresh, the goal is to explore the challenge of letter frequency in particular, and patterns in particular, for simple encryption algorithms.

1. What are the frequencies of the letters in the plaintext? Write a C program that reads in text from a file into a buffer, counts the occurrences of each [lowercase] letter of the English alphabet, and computes the relative frequencies. Your program should print out the contents of the buffer, and the frequency results in a simple list such as the one above.
2. Add a function to encrypt the plaintext with a Vigenère cipher and a given key. You should add the key as a command line argument so that you can enter a different key each time you run. A key is a text string of max length 4, min length 1.
3. Add the functionality to count the occurrences of each [lowercase] letter of the English alphabet in the ciphertext, compute the relative frequencies, and print out the ciphertext and the frequency

results in a simple list.

4. Run your encryption program over the plaintext for two different keys: yz and wxyz.
5. Submit a table of your results, First column is the alphabet, second is the relative frequency of each, 3rd column is frequency from plaintext, 4th column is frequency from key yz, and 5th column is frequency from key wxyz.
6. what happens to your program if the text in your file is too long to fit in the buffer? If you try to enter a key with length 5?

Response: Vigenère Ciphers output:

Letter	Plaintext:	Key – "yz":	Key – "wxyz":
A:	064 7.273%	041 4.659%	039 4.432%
B:	011 1.250%	019 2.159%	050 5.682%
C:	037 4.205%	039 4.432%	045 5.114%
D:	031 3.523%	077 8.750%	043 4.886%
E:	109 12.386%	054 6.136%	042 4.773%
F:	015 1.705%	010 1.136%	026 2.955%
G:	013 1.477%	023 2.614%	029 3.295%
H:	030 3.409%	045 5.114%	023 2.614%
I:	068 7.727%	042 4.773%	036 4.091%
J:	002 0.227%	002 0.227%	014 1.591%
K:	006 0.682%	021 2.386%	034 3.864%
L:	037 4.205%	029 3.295%	045 5.114%
M:	018 2.045%	047 5.341%	036 4.091%
N:	061 6.932%	054 6.136%	029 3.295%
O:	056 6.364%	036 4.091%	040 4.545%
P:	021 2.386%	011 1.250%	030 3.409%
Q:	003 0.341%	030 3.409%	052 5.909%
R:	054 6.136%	054 6.136%	059 6.705%
S:	064 7.273%	079 8.977%	057 6.477%
T:	077 8.750%	054 6.136%	031 3.523%
U:	044 5.000%	032 3.636%	017 1.932%
V:	012 1.364%	012 1.364%	012 1.364%
W:	011 1.250%	003 0.341%	008 0.909%
X:	001 0.114%	011 1.250%	021 2.386%
Y:	022 2.500%	012 1.364%	024 2.727%
Z:	001 0.114%	031 3.523%	026 2.955%