Homework #1

CS468 Secure Programming and Systems Fall 2021

This homework is due on 9/17/2021 by the end of class. This is an individual homework assignment to be done by each student individually. The submission of your homework is your acknowledgement of the honor code statement

> *I have not copied any solution from anyone and not provided any solution to anyone on this assignment. The solution has been entirely worked out by me and represents my individual effort.*

Please

- Use a word processor (MS-Word, LaTex, Framemaker, …) for your solutions. No hand writing submission will be accepted.
- Include your name and G-number at the beginning of your homework submission.
- Pack all your files with zip, tar or gzip etc. and name your packed file as CS468_HW1_<your last name>_<your G#>.zip (or tar.gz)
- Submit your packed solution to the blackboard

1. Textbook Problems 2.1 at page 56 (15 points)

   A generalization of the Caesar cipher, known as the affine Caesar cipher, has the following form: For each plaintext letter p, substitute the ciphertext letter C:

   $$C = E([a, b], p) = (ap + b) \bmod 26$$

   A basic requirement of any encryption algorithm is that it be one-to-one. That is, if $p \neq q$, then $E(k, p) \neq E(k, q)$. Otherwise, decryption is impossible, because more than one plaintext character maps into the same ciphertext character. The affine Caesar cipher is not one-to-one for all values of a. For example, for $a = 2$ and $b = 3$, then $E([a, b], 0) = E([a, b], 13) = 3$.

   a. Are there any limitations on the value of b? Explain why or why not.
   b. Determine which values of a are not allowed.
   c. Provide a general statement of which values of a are and are not allowed. Justify your statement.

2. Textbook Problems 2.3 at page 58 (15 points)

   A ciphertext has been generated with an affine cipher. The most frequent letter of the ciphertext is “B,” and the second most frequent letter of the ciphertext is “U.” Break this code by figuring out a and b of the affine cipher.

   Hint: Check out Figure 2.5 for English letter frequency.

3. Textbook Problems 2.10 at page 58 (20 points)

   a. Construct a Playfair matrix with the key *largest*.
   b. Construct a Playfair matrix with the key *occurrence*. Make a reasonable assumption about how to treat redundant letters in the key.

4. Playfair Cipher Encryption (15 points)

Using this Playfair matrix:

M	F	H	I/J	K
U	N	O	P	Q
Z	V	W	X	Y
E	L	A	R	G
D	S	T	B	C

a. Encrypt this message:

You should be able to encrypt this sentence

Note: Here we only consider letters when encrypting/decrypting.

b. Repeat part (a) using the Playfair matrix from Problem 2.10a (problem 3.a).

c. How do you explain the relationship between the results of problem a and problem b? Can you generalize your conclusion?

5. Textbook Problems 2.20 at page 59 (15 points)

This problem explores the use of a one-time pad version of the Vigenère cipher. In this scheme, the key is a stream of random numbers between 0 and 26. For example, if the key is 3 19 5 ..., then the first letter of plaintext is encrypted with a shift of 3 letters, the second with a shift of 19 letters, the third with a shift of 5 letters, and so on.

a. Encrypt the plaintext sendmoremoney with the key stream:
9 0 1 7 23 15 21 14 11 11 2 8 9

b. Using the ciphertext produced in part (a), find a key so that the ciphertext decrypts to the plaintext cashnotneeded

6. Textbook Problems 3.11 at page 83 (20 points)

This problem provides a numerical example of encryption using a one-round version of DES. We start with the same bit pattern for the key *K* and the plaintext, namely:
Hexadecimal notation: 0 1 2 3 4 5 6 7 8 9 A B C D E F
Binary notation: 0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111

a. Derive $K1$, the first-round subkey.
b. Derive $L0$, $R0$.
c. Expand $R0$ to get E[$R0$], where E[.] is the expansion function of Table S.1.
d. Calculate $A = \mathrm{E}[R0] \oplus K1$.
e. Group the 48-bit result of (d) into sets of 6 bits and evaluate the corresponding S-box substitutions.
f. Concatenate the results of (e) to get a 32-bit result, B.

g. Apply the permutation to get P(B).

h. Calculate $R1$ = P(B) $\bigoplus$ $L0$.

i. Write down the ciphertext.