Homework #1
Chap. 1, 3, 4

All Homework exercises are either hand written or programming

Homework #1

- Chap.1:
  - Review question 1.3
- Chap.3:
  - Review question 3.4, 3.12
  - Problem: 3.1, 3.3
  - Programming problem: 3.23*, 3.25*
- Chap.4:
  - Review question 4.4, 4.6
  - Problem: 4.2, 4.6
  - Programming problem: 4.20*
- Due: Feb 06, 2022

- Chap. 1
  - Review question 1.3: List and briefly define categories of passive and active attacks.
- Chap. 3
  - Review question 3.4: What is the difference between a block cipher and a stream cipher?
  - Review question 3.12: What are the two problems with the one-time pad?

(pg 89)

(pg 106)

– Problem 3.1: 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!=q, then E(k,p)!=E(k,q). Otherwise, decryption is impossible, because more than one plaintext character maps into the same ciphertext character.

(…to be continued)

(…continued from the previous slide)

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.

– Problem 3.3: A ciphertext has been generated with an affine cipher. The most frequent letter of the ciphertext is “C,” and the second most frequent letter of the ciphertext is “Z.” Break this code.

https://www.thecrazyprogrammer.com/2016/11/caesar-cipher-c-c-encryption-decryption.html

https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/

- Programming Problem 3.23: Write a program that can encrypt and decrypt using the general Caesar cipher, also known as an additive cipher.
- Programming Problem 3.25: Write a program that can perform a letter frequency attack on an additive cipher without human intervention. Your software should produce possible plaintexts in rough order of likelihood. It would be good if your user interface allowed the user to specify “give me the top 10 possible plaintexts.”

- Chap. 4
  - Review question 4.4: Briefly define the terms substitution and permutation.
  - Review question 4.6: Which parameters and design choices determine the actual algorithm of a Feistel cipher?

– Problem 4.2: Consider a Feistel cipher composed of sixteen rounds with a block length of 128 bits and a key length of 128 bits. Suppose that, for a given k, the key scheduling algorithm determines values for the first eight round keys, k1, k2, …, k8, and then sets
k9=k8, k10=k7, k11=k6, …, k16=k1
(To be continued…)

- (…continued from the previous slide)
- Suppose you have a ciphertext c. Explain how, with access to an encryption oracle, you can decrypt c and determine m using just a single oracle query. This shows that such a cipher is vulnerable to a chosen plaintext attack.
  (To be continued….)

– (…continued from the previous slide)
(An encryption oracle can be thought of as a device that, when given a plaintext, returns the corresponding ciphertext. The internal details of the device are not known to you and you cannot break open the device. You can only gain information from the oracle by making queries to it and observing its responses.)

– Problem 4.6: Suppose the DES F function mapped every 32-bit input R, regardless of the value of the input K, to:
  (a) 32-bit string of zero
  (b) R
  Then,
  1. What function would DES then compute?
  2. What would the decryption look like?

  Hint: Use the following properties of the XOR operation:
  $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
  $(A \oplus A) = 0$, $(A \oplus 0) = A$
  $A \oplus 1$ = bitwise complement of A
  where A, B, C are n-bit strings of bits
  0 is an n-bit string of zeros,
  1 is an n-bit string of ones.

- Programming Problem 4.20: Create software that can encrypt and decrypt using S-DES, described in Appendix G. Test data: use plaintext, ciphertext, and key of Problem 4.18.

- Note: The ciphertext, and key of Problem 4.18 are: 01000110, 1010000010.

Homework Submission

- For hand-written exercises, please scan, convert to compressed pdf and submit to our homework submission site.
- For programming exercises, please submit to our homework submission site.
  - Program uploading: a compressed file (in .zip format) including source codes and compilation instructions if it needs special environment to compile or run
    - Please clearly name your program files using your ID
  - *Note: the uploaded file size must be less than 5MB*