Micah Too

Assignment 2

Chapter 3

1. (5 points) Complete Problem 2 (a, b) from the text.
   This problem deals with stream ciphers.
   a. If we generate a sufficiently long keystream, the keystream must eventually repeat. Why?
   We eventually repeat because the algorithm to generate keystream is deterministic. Using the same initial Value and key to generate will eventually give the same sequence of bits. The keystream is generated by a finite set of operations hence the output will be repeated at some point

   b. Why is it a security concern if the keystream repeats?

   If period of repetition is known, attacker can carry out a chosen ciphertext attack or chosen plaintext attack and from there can overlap the periods or XoR them to get a distinguishable pattern of the keystream. Once keystream is known attacker can encrypt and decrypt any other ciphertext and plaintexts

2. (10 points) Suppose that Alice uses a stream cipher to encrypt plaintext P, obtaining ciphertext C, and Alice then sends C to Bob. Suppose that Trudy happens to know the plaintext P, but Trudy does not know the key K that was used in the stream cipher.
   a. Show that Trudy can easily determine the keystream that was used to encrypt P.
   if Trudy knows the plaintext P and ciphertext C, he can compute the key by P xor C which results in the same result as doing C xor K
   since C = P xor K then to get K = P xor C
   b. Show that Trudy can, in effect, replace P with plaintext of her choosing, say, P'. That is, show that Trudy can create a ciphertext message C'' so that when Bob decrypts C' he will obtain P'.
   since Trudy knows the key K, he can replace C with
   since Trudy knows the keystream used for encryption P to get C`` he can XOR K with P`
   during decryption Bob will obtain Trudy's replaced plaintext P` because to decrypt C``, he will do K xor C`` which gives P`

   P =P` according to Trudy replaced plaintext
   C`` = P` xor K encrypting
   Decrypting
   K xor C`` = P`

3. (15 points) This problem deals with the A5/1 cipher. For each part, justify your answer.
   a. On average, how often does the X register step?
   b. On average, how often does the Y register step?
   c. On average, how often does the Z register step?
   d. On average, how often do all three registers step?
   e. On average, how often do exactly two registers step?
   f. On average, how often does exactly one register step?
   g. On average, how often does no register step?

A5/1 cipher.

X_8	Y_{10}	Z_{10}	maj(X_8, Y_{10}, Z_{10})	X_{st}	Y_{st}	Z_{st}
0	0	0	0	1	1	1
0	0	1	0	1	1	0
0	1	0	0	1	0	1
0	1	1	1	0	1	1
1	0	0	0	0	1	1
1	0	1	1	1	0	1
1	1	0	0	1	1	0
1	1	1	1	1	1	1

a) X register (the 1's)
steps $\frac{6}{8}$ times $= 0.75$

b) Y register $= \frac{6}{8}$ times $= 0.75$

c) Z register $= \frac{6}{8}$ times $= 0.75$

d) all 3 registers
last 2 combinations (the 0's)
$= \frac{2}{8} = 0.25$

e) two registers steps
$X4 = \frac{4}{8}$
$YZ = \frac{2}{8} = \frac{1}{8}$.
$ZX = \frac{1}{8}$

f) only 1 register
$6/8 = \frac{3}{4}$.

g) there is no step

4. (5 points) This problem deals with a Feistel Cipher.
   a. Give the definition of a Feistel Cipher.
      A type of block cipher that iteratively applies round function to half of the block and XORs the output with the second half of the block then swaps them and repeats the process for a fixed number of rounds. It's named after German physicist Horst Feistel.
   b. Is DES a Feistel Cipher?
      Yes, Data Encryption Standard is a Feistel cipher that uses 16 round Feistel Structure to encrypt 64 bit block of plaintext

   c. Is AES a Feistel Cipher?
      AES Advanced Encryption Standard is not a Feistel cipher. It uses a substitution-permutation method network.
   d. Why is the Tiny Encryption Algorithm, TEA, "almost" a Feistel Cipher?
      Tiny Encryption Algorithm uses a similar structure as the Feistel cipher by iteratively applying the round function to half the block hence the need for separate encryption and decryption routines. It also uses addition and subtraction instead of XOR.

5. (5 points) Suppose that we instead define double DES as $C = D(E(P,K_1),K_2)$. Describe a meet-in-the-middle attack on this cipher.
   Compute a table of possible values for intermediate ciphertext resulting from encrypting known plaintext with K1 and K2
   Compute corresponding intermediate ciphertext using k1 and K2 for each value resulting from decrypting known ciphertext with K2
   Check if it exists in table in step 1, for all matches store corresponding pairs of intermediate plaintexts and ciphertext
   Compare list of plaintext and ciphertext pairs from step 2 with table in step 1. For matches, the corresponding K1 and K2 are the keys.
6. (10 points) Recall the meet-in-the-middle attack on double DES discussed in this chapter. Assuming that chosen plaintext is available, this attack recovers a 112-bit key with about the same work needed for an exhaustive search to recover a 56-bit key, that is, about 2^{55}.

   a. If we only have known plaintext available, not chosen plaintext, what changes do we need to make to the double DES attack?
   If we have known plaintext, attacker cannot choose plaintext to encrypt, they can intercept the ciphertext and use it as an already known output.
   First, attacker generates a table of possible values of $A(P, K_1)$ for possible K_1 then computes $D(C, K_2)$ for each of them and store pair of $(E(P, K_1)D(C, K_2)$, sort and

search table for matching values of the two. For every match, attacker computes C` = E(P, $K_1$) and P` =D(C`, K_2)
If resulting P` matches known plaintext, then K1 and K2 are correct keys.
b. What is the work factor for the known plaintext version of the meet-in-the-middle double DES attack?
Work factor
First encryption K1, = 2^56,
Second encryption K2 = 2^56
Search table for all possible matches
Total no of operations = (2^56) +(2^56) = 2^57

Effective key size double des 112 bits, = (2^112)
The meet in the middle attack reduces the work factor from 2^112 to 2^57 which is a factor of (1/ 2^{55})

7. (10 points) Recall that an initialization vector (IV) need not be secret.
   a. Does an IV need to be random?
      Yes it need to be random so that identical plaintexts do not result in identical ciphertexts during encryption
   b. Discuss possible security disadvantages (or advantages) if IVs are selected in sequence instead of being generated at random.
      Predictability – attacker may be able to predict the next IV in the sequence by analyzing the previous IV compromising the security of the encryptions
      Reuse – if IV is used more than once it can compromise the security of an encryption scheme.
      Simple implementation of encryption because there wont be need method for generating the sequence IVs

8. (10 points) The formula for counter mode encryption is $C_i = P_i$ xor E(IV + i, K). Suppose instead we use the $C_i = P_i$ xor E(K, IV + i). Is this secure? If so, why? If not, why not?
   No, its not secure. Same IV is used for two different messages encrypted with the same key. An attacker can get the XOR of two plaintexts or two ciphertexts to get the key.

9. (10 points) Suppose that we use a block cipher to encrypt according to the rule C0 = IV xor (P0, K), C1 = C0 xor E(P1 K), C2 = C1 xor E(P2, K), ... Download more at Learnclax.com 84 SYMMETRIC KEY CRYPTO
   a. What is the corresponding decryption rule?
      Corresponding decryption
      P_0 = IV xor(C_0, K),
      $P_1 = P_0$ xor E(C_1,K)

$P_2 = P_1$ xor $E(C_2, K)$……

b. Give two security disadvantages of this mode as compared to CBC mode.
Each plaintext block is XOR with same keystream value, hence same plaintext block will always encrypt to same ciphertext block and can be exploited by attackers to do a known plaintext attack.
Any error that occurs in the ciphertext will affect the decryption of all the other blocks because each ciphertext is used a s a input to encrypting the next block.

10. (10 points) Suppose that Alice and Bob decide to always use the same IV instead of choosing IVs at random.
a. Discuss a security problem this creates if CBC mode is used.
Using same IV makes it vulnerable to a chosen-plaintext attack. The attacker can learn information about the plaintext and possibly recover the encryption key by encrypting a plaintext and observing the resulting ciphertext

b. Discuss a security problem this creates if CTR mode is used.
Using the same IV always will make the CTR mode vulnerable to known=plaintext attack. Attacker observes produced cyphertext from encrypting known plaintext with fixed IV and can use the information to recover the keystream used during encryption allowing decryption of other ciphertext that used similar IV

c. If the same IV is always used, which is more secure, CBC or CTR mode?
CTR mode is more secure than the CBC mode when using a fixed IV because it does not depend on the previous block of ciphertext for encryption

11. (10 points) Assume a particular Feistel cipher uses the round function $F(X, K) = X \oplus K$, and number of rounds n = 4. Let the plaintext block P be the 8-bit binary number 10110101, and the subkeys K1 through K4 as follows: 1011, 0100, 0101, 1010. Run the cipher on this input, and show the values of Li and Ri for each round i, as well as the final ciphertext block that is obtained. You do not have to compute each step by hand — you may write a simple program which gives the required outputs. If you choose to write a program, please submit your source code, and README file, along with the rest of the assignment as a .zip file.

Apply Feistel cipher with round function

$F(X, K) = X \oplus K$

$n = 4$

Round 1:

$L_0 = 101101$

$R_0 = 010101$

$L_1 = R_0 = 010101$

$R_1 = L_0 \oplus F(R_0, K_1)$

$F(R_0, K_1) = (101101) \text{ XOR } (010101 \text{ xor } 1011)$

181 21 11

XOR 0 1 0 1 0 1 XOR 1 0 1 1 0 1

XOR 0 0 1 0 1 1 0 1 1 1 1 0

0 1 1 1 1 0 1 1 1 1 1 1

30

Round 2:

$L_1 = 010101$, $R_1 = 111111$

$L_2 = R_1 = 111111$

$R_2 = L_1 \oplus F(R_1, K_2) = 010101 \text{ xor } (111111 \oplus 0100)$

$= 101010$

Round 3.

$L_2 = 111111$, $R_2 = 101010$

$L_3 = R_2 = 101010$

$R_3 = L_2 \oplus F(R_2, K_3) = 111111 \oplus (101010 \oplus 0101)$

$= 000100$

Round 4

$L_3 = 101010$, $R_3 = 000100$

$L_4 = R_3 = 000100$

$R_4 = L_3 \oplus F(R_3, K_4) = 101010 \text{ xor } (0001000 \text{ xor } 1010)$

$= 101100.$

Final cipher $C = R_4 \| L_4 = 000100 \| 000100$ (concatenating)

$= 10110000$