**Solutions for Assignment 2**

Rahul Chowdary Garigipati

Chapter-3:

1. This problem deals with stream ciphers.
2. If we generate a sufficiently long keystream, the keystream must eventually repeat. Why?

Ans) In stream ciphers, stream cipher takes a key ‘K’ of n - bits in length and stretches it into a long keystream, as to make it to the length of a plaintext as the key size is fixed. The keystream must eventually repeat by padding bits until it matches the length of the plain text, and the certain part of the keystream gets eventually repeated.

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

Ans) we are stretching a keystream to match the length of the plain text to perform the **XOR** operation. The problem with this is that repeating keystream would also give us a repeating pattern of a cipher text and it would be easier for the intruder to interpret the data and guess the key.

1. 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.
2. Show that Trudy can easily determine the keystream that was used to encrypt P.

Ans) A stream cipher uses an **XOR** operation to encrypt plain text to cipher texts. Now Turdy knows the plain text and the cipher text then she has to determine the keystream. She can simply **XOR** the plain text with the cipher text to get the key.

Consider the plain text Pi and a cipher text Ci. We know that C0 = P0 ⨁ K0; C1 = P1 ⨁ K1; …..; Cn = Pn ⨁ Kn. If Trudy knows even one single plain text or cipher text pair, then she can recover the keystream Ki = Ci ⨁ Pi.

1. 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'.*

Ans) Now Trudy can replace the plaintext by her wish, say, P*'* and the keystream is already known. She can generate a new ciphertext, C*'* by doing an **XOR** between the keystream and the plain text and send, such that when Bob decrypts C*'* he will obtain P*'*.

1. This problem deals with the A5/1 cipher. For each part, justify your answer.
2. On average, how often does the *X* register step?

Ans) Consider, at each step: *m* = maj(*x*8, *y*10, *z*10)

Examples: maj(0,1,0) = 0 and maj(1,1,0) = 1

*x*8, *y*10, *z*10,  Maj(*x*8, *y*10, *z*10), X step, Y step, Z step

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 1 1 1 0

1 1 1 1 1 1 1

If *x*8 = *m* **then X steps**,

* *t* = *x*13 ⨁ *x*16 ⨁ *x*17 ⨁ *x*18
* *xi* = *xi-1* for *i* = 18,17,…,1 and *x*0 = *t*

Therefore, on average X register steps 6/8 times (or) 0.75 times.

1. On average, how often does the *Y* register step?

Ans) If *y*10 = *m***then Y steps**,

* *t* = *y*20 ⨁ *y*21
* *yi* = *yi-1* for *i* = 21,20,…,1 and *y0 = t*

Therefore, on average Y register steps 6/8 times (or) 0.75 times.

1. On average, how often does the *Z* register step?

Ans) If *z*10 = *m* **then Z steps,**

* *t =* z7 ⨁ *z*20 ⨁ *z*21 ⨁ *z*22
* *zi* = *zi-1* for *i* = 22,21,…,1 and *z*0 = *t*

Therefore, on average Z register steps 6/8 times (or) 0.75 times.

1. On average, how often do all three registers step?

Ans) On average, all 3 registers steps 2/8 times (or) 0.25 times.

1. On average, how often do exactly two registers step?

Ans) On average, two registers steps exactly 6/8 times (or) 0.75 times.

1. On average, how often does exactly one register step?

Ans) One register doesn`t step.

1. On average, how often does no register step?

Ans) None.

1. This problem deals with a Feistel Cipher.
2. Give the definition of a Feistel Cipher.

Ans) **Feistel cipher** refers to a type of block cipher design. In a Feistel cipher, the plaintext block is split into left and right halves, P=(L0,R0), and for each round i = 1,2,... ,n. Now left and right halves are computed according to the rules Li = Ri-1,

Ri = Li-1 ⨁ F(Ri-1, Ki). Where ‘F’ is **round function** and ‘Ki’ is **subkey**. The ciphertext C is the output of the final round, namely, C = (Ln, Rn).

1. Is DES a Feistel Cipher?

Ans) DES is a Feistel Cipher.

1. Is AES a Feistel Cipher?

Ans) AES is not a Feistel Cipher.

1. Why is the Tiny Encryption Algorithm, TEA, "almost" a Feistel Cipher?

Ans) As DES uses **XOR** operation it is a Feistel Cipher. Whereas TEA uses **Addition** and **Subtraction** instead of **XOR**. So, the number of rounds for the TEA is more like two rounds of DES and hence TEA is not a Feistel Cipher.

1. Recall the attack on double DES discussed in the text. Suppose that we instead define double DES as C = D(E(P, K1), K2). Describe a meet-in-the-middle attack on this cipher.

Ans) The intruder will build a table that will have two columns, where the first column will be for the encryption of plane texts E(P, ), …., E(P, ). And the second column will be for the encryption of the cipher texts E(C, ), ….., E(C, ). There will be a match at some point and the intruder can identify K1, and K2.

1. 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 255.
2. If we only have known plaintext available, not chosen plaintext, what changes do we need to make to the double DES attack?

Ans) The attack will be the same except that we need to compute the lookup table each time we conduct the attack.

1. What is the work factor for the known plaintext version of the meet-in-the-middle double DES attack?

Ans) We must include the work to compute the lookup table, This gives us a work factor of **(256) + (255).**

1. Recall that an initialization vector (IV) need not be secret.
2. Does an IV need to be random?

Ans) An IV needs to be random as it provides randomization.

1. Discuss possible security disadvantages (or advantages) if IVs are selected in sequence instead of being generated at random.

Ans) An IV should never be chosen in a predictable way as in a sequence, since it will enable an intruder to perform a chosen plaintext attack. For example, If 2 IVs are the same for the same plaintext, then the resulting ciphertext will also be the same.

1. The formula for counter mode encryption is Ci = Pi ⨁ E(IV + i, K). Suppose instead we use the formula Ci = Pi ⨁ E(K, IV + i). Is this secure? If so, why? If not, why not?

Ans) No it is not secure, as every time a counter initiated value is encrypted with that of the key and is XOR`ed with the plaintext, which results in the ciphertext. It generates the next keystream by encrypting successive values, the keystream is then stretched to match the length of the plaintext. The problem with this is that repeating keystream would also give us a repeating pattern of a cipher text and it would be easier for the intruder to interpret the data and guess the key.

1. Suppose that we use a block cipher to encrypt according to the rule *C0* = IV ⨁ *E(P0, K), C1 = C0* ⨁  *E(P1, K), C2 = C1* ⨁ *E(P2, K), …*
2. What is the corresponding decryption rule?

Ans) Decryption rule:

P0 = D(C0 ⨁ IV, K), P1 = D(C1 ⨁ C0, K), P2 = D(C2 ⨁ C1, K),…., Pn = D(Cn ⨁ Cn-1, K).

1. Give two security disadvantages of this mode as compared to CBC mode.

Ans) The disadvantages when compared to the CBC mode is that, if one bit of the cipher text is tampered, that would result in the loss of all the subsequent blocks of data. every block of ciphertext depends on the previous block of ciphertext, rather than the previous block of plaintext.

1. Suppose that Alice and Bob decide to always use the same IV instead of choosing IVs at random.
2. Discuss a security problem this creates if CBC mode is used.

Ans) If the same IVs are used in the CBC for the same plaintext, then the resulting ciphertext will also be the same. Hence it will enable an intruder to perform a chosen plaintext attack and can break the key.

1. Discuss a security problem this creates if CTR mode is used.

Ans) If the same IVs are used in CTR for encrypting different messages, there could be a chance that different plaintexts get encrypted using the same key. Therefore, by doing an **XOR** operation for **plaintexts** and the **ciphertexts** can give the intruder the key. So, this would be very easy for the intruder to get to know the plain texts once he knows the IVs and Key and Ciphertexts.

1. If the same IV is always used, which is more secure, CBC or CTR mode?

Ans) CBC is somewhat better as two plain texts needs to be identical for the intruder to succeed. Whereas, In CTR, attacker can succeed even if two plain texts are different.

1. Assume a particular Feistel cipher uses the round function F(X, K) = X ⨁ 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.

Ans) Given Plain Text: 10110101 and the subkeys from K1 to K4 as follows:

K1=1011

K2=0100

K3=0101

K4=1010

We know that round function F (X, K) = X ⊕ K.

Dividing the given plain text into two equal halves (L0, R0) = (1011, 0101)

In Feistel cipher L1=R0 and R1= L0 ⊕ F (R0, key1)

**Round1:**

So L1= 0101 and R1=1011 ⊕ (0101 ⊕ 1011)

Hence R1=0101.

**Round2:**

Now, L2=R1

Hence L2=0101

R2= L1 ⊕ F (R1, key2)

R2=0101 ⊕ (0101⊕0100)

R2=0100.

**Round3:**

Now, L3=R2

HenceL3=0100

R3= L2 ⊕ F (R2, key3)

R3=0101 ⊕ (0100 ⊕ 0101)

R3=0100.

**Round4:**

Now, L4=R3

Hence L4=0100

R4= L3 ⊕ F (R3, key4)

R4=0100 ⊕ (0100 ⊕ 1010)

R4=1010.

Therefore,Ciphertext = (L4, R4) **= (01001010)**.