Assignment 3

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

Stream cipher uses similar logic to the “one-time pad” logic, therefore if we know the plaintext P and the ciphertext C, we can know the keystream K by doing C xor P.

**K = C ⊕ P**

* 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'.

Since Trudy knows the plaintext P, she can determine the keystream K by doing C xor P and then use the keystream to encrypt another plaintext P’ by doing P’ xor K, therefore creating ciphertext C’

**C’ = C ⊕ P ⊕ P’ = K ⊕ P’**

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

In order for X to register step, X must have the same value than Y or Z or both. Since the values are only 0 and 1, the chance that X has the same value as Y is 50%, and if X is not the same as Y, the chance that X is the same value as Z is also 50%, making it a total of 75% chance for X to register a step.

**75% or 75 times out of 100**

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

In order for Y to register step, Y must have the same value than X or Z or both. Since the values are only 0 and 1, the chance that Y has the same value as X is 50%, and if Y is not the same as X, the chance that Y is the same value as Z is also 50%, making it a total of 75% chance for Y to register a step. **75% or 75 times out of 100**

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

In order for Z to register step, Z must have the same value than X or Y or both. Since the values are only 0 and 1, the chance that Z has the same value as X is 50%, and if Z is not the same as X, the chance that Z is the same value as Y is also 50%, making it a total of 75% chance for Z to register a step. **75% or 75 times out of 100**

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

Since we have three LFSRs X Y and Z, and their value can only be 0 or 1, this implies that X Y and Z can only have 23 different combinations. But in order for all three register to steps, there can only be 2 combinations that satisfy the requirement, either all 0s or all 1s. Therefore it is 2 out of 23.

**2/23 = 2/8 = 0.25, 25% or 25 times out of 100.**

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

The stream cipher is designed so that at least 2 registers will step every time, and from out previous question, there are 23 combinations and only 2 of them can cause all three registers to step. Therefore, there are 6 combinations that will exactly cause two registers to step.

**6/23 = 6/8 = 0.75, 75% or 75 times out of 100.**

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

In a stream cipher, at least two registers will step every time, therefore it can **NEVER** step only one register. **0%**

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

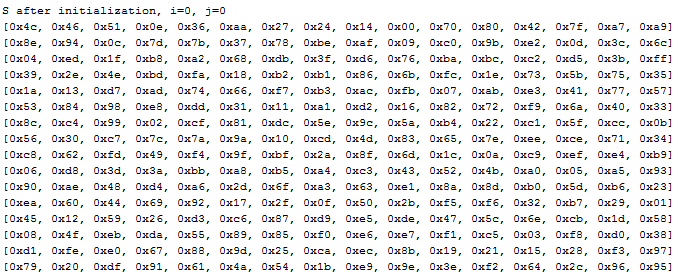
In a stream cipher, at least two registers will step every time, therefore the possibility of no register to step is **NONE, 0%.**

1. **For bits x, y, and z, the function maj(x, y, z) is defined to be the majority vote, that is, if two or more of the three bits are 0, then the function returns 0; otherwise, it returns l. Write the truth table for this function and derive the boolean function that is equivalent to maj(x, y, z).**

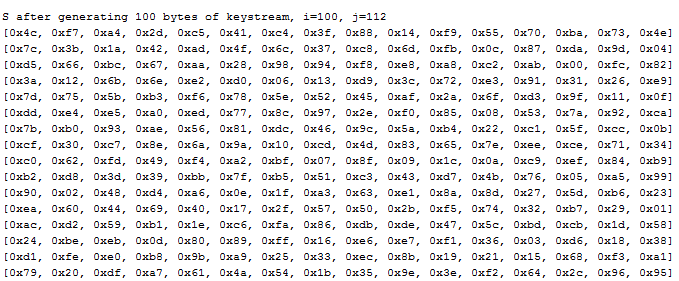
|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | Z | maj |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |

Boolean function: **maj = (X AND Y) OR (X AND Z) OR (Y AND Z) = XY+XZ+YZ**

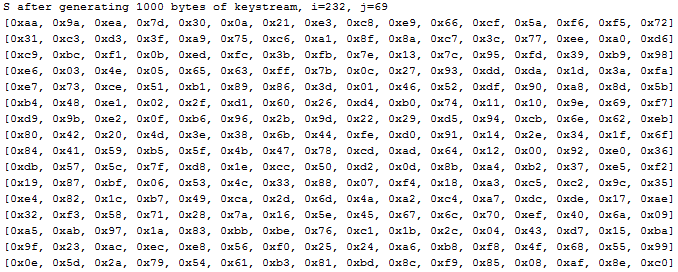
1. **Implement the RC4 algorithm. Suppose the key consists of the following seven bytes: (0x1A, 0x2B, 0x3C, 0x4D, 0x5E, 0x6F, 0x77). For each of the following, give S in the form of a 16 x 16 array where each entry is in hex.**
   1. List the permutation S and indices *i* and *j* after the initialization phase has completed.



* 1. List the permutation S and indices *i* and *j* after the first 100 bytes of keystream have been generated.



* 1. List the permutation S and indices *i* and *j* after the first 1000 bytes of keystream have been generated.



1. **Consider a Feistel cipher with four rounds. Then the plaintext is denoted as P =(L0, R0) and the corresponding ciphertext is C = (L4, R4). What is the ciphertext C, in terms of L0, R0, and the subkey, for each of the following round functions?**
   1. *F*(Ri-1,Ki) = 0

L1 = R0 , R1 = L0 ⊕ *F*(Ri-1,Ki) = L0 ⊕ 0 = L0

L2 = R1 = L0 , R2 = L1 ⊕ *F*(Ri-1,Ki) = R0 ⊕ 0 = R0

L3 = R2 = R0 , R3 = L2 ⊕ 0 = L2 = L0

L4 = R3 = L0 , R4 = L3 ⊕ 0 =R0

**C = (L0, R0)**

* 1. *F*(Ri-1,Ki) = Ri-1

L1 = R0 , R1 = L0 ⊕ *F*(Ri-1,Ki) = L0 ⊕ R0

L2 = R1 = L0 ⊕ R0 , R2 = L1 ⊕ *F*(Ri-1,Ki) = R0 ⊕ R1 = R0 ⊕L0 ⊕ R0 = L0

L3 = R2 = L0 , R3 = L2 ⊕ *F*(Ri-1,Ki) = L2 ⊕ R2 = L0 ⊕ R0 ⊕ L0 = R0

L4 = R3 = R0 , R4 = L3 ⊕ *F*(Ri-1,Ki) = L0 ⊕ R3 = L0 ⊕ R0

**C = (R0, L0 ⊕ R0)**

* 1. *F*(Ri-1,Ki) = Ki

L1 = R0 , R1 = L0 ⊕ *F*(Ri-1,Ki) = L0 ⊕ K1

L2 = R1 = L0 ⊕ K1 , R2 = L1 ⊕ *F*(Ri-1,Ki) = R0 ⊕ K2

L3 = R2 = R0 ⊕ K2 , R3 = L2 ⊕ *F*(Ri-1,Ki) = L0 ⊕ K1 ⊕ K3

L4 = R3 = L0 ⊕ K1 ⊕ K3, R4 = L3 ⊕ *F*(Ri-1,Ki) = R0 ⊕ K2 ⊕ K4

**C = (L0 ⊕ K1 ⊕ K3 , R0 ⊕ K2 ⊕ K4)**

* 1. *F*(Ri-1,Ki) = Ri-1 ⊕Ki

L1 = R0 , R1 = L0 ⊕ *F*(Ri-1,Ki) = L0 ⊕ K1 ⊕ R0

L2 = R1 = L0 ⊕ K1 ⊕ R0 , R2 = L1 ⊕ *F*(Ri-1,Ki) = R0 ⊕ K2 ⊕ R1 = R0 ⊕ K2 ⊕ L0 ⊕ K1 ⊕ R0 = K2 ⊕ L0 ⊕ K1

L3 = R2 = K2 ⊕ L0 ⊕ K1 , R3 = L2 ⊕ *F*(Ri-1,Ki) = L0 ⊕ K1 ⊕ R0 ⊕ K3 ⊕ R2 = L0 ⊕ K1 ⊕ R0 ⊕ K3 ⊕ K2 ⊕ L0 ⊕ K1 = R0 ⊕ K3 ⊕ K2

L4 = R3 = R0 ⊕ K3 ⊕ K2 , R4 = L3 ⊕ *F*(Ri-1,Ki) = K2 ⊕ L0 ⊕ K1 ⊕ K4 ⊕ R3 = K2 ⊕ L0 ⊕ K1 ⊕ K4 ⊕ R0 ⊕ K3 ⊕ K2 = L0 ⊕ K1 ⊕ K4 ⊕ R0 ⊕ K3

**C = (R0 ⊕ K3 ⊕ K2, L0 ⊕ K1 ⊕ K4 ⊕ R0 ⊕ K3)**

1. **This problem deals with the DES cipher.**
   1. How many bits in each plaintext block?

DES has 64bits in each plaintext block

* 1. How many bits in each ciphertext block?

DES has 64bits in each ciphertext block.

* 1. How many bits in the key?

DES has a key of 56 bits

* 1. How many bits in each subkey?

In DES, 48 bits keys are used in each round, also known as Subkey

* 1. How many rounds?

There are 16 rounds in DES

* 1. How many S-boxes?

There are 8 S-Boxes in DES

* 1. An S-box requires how many bits of input?

An S-Box requires 6 bits of input

* 1. An S-box generates how many bits of output?

An S-Box generates 4 bits of output

1. **AES consists of four functions in three layers.**
   1. Which of the four functions are primarily for confusion and which are primarily for diffusion? Justify your answer.

ByteSub and AddRoundKey are confusion, because they substitute the overall content, shiftrow and mix column are diffusion because they move content around.

* 1. Which of the three layers are for confusion and which are for diffusion? Justify your answer.

Linear mixing layers are diffusion, because it swaps/mix the content around. Non-Linear layer and key addition layer are confusion because it simply substitutes the content with other characters as a whole.

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), …**

1. What is the corresponding decryption rule?

P0 = D(IV ⊕ C0, K)

P1 = D(C0 ⊕ C1, K)

P2 = D(C2 ⊕ C1, K),…

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

To have an IV to XOR the encrypted message is useless, because the IV is not a secret number, and if intruders know the IV, they can simply remove it from the ciphertext by performing XOR IV on it.

The ciphertext blocks are also not dependent of the other ones, for example C1 = C0 ⊕ E(P1,K), but since we know C0 we can just perform XOR on C1 and we are left with C1 = E(P1,K)

1. **Explain how to do random access on data encrypted in CBC mode. Are there any significant disadvantages of using CBC mode for random access as compared to CTR mode?**

To perform random access on CBC mode, we have to obtain the block we wish to decrypt and also the block that precedes it, to get the plaintext, first we need to decrypt it with the key and then perform XOR on it with the preceding block.

One of the disadvantages of using CBC mode for random access, is that random access on CBC mode requires the preceding block in order to be able to get the plaintext, compared to CTR, we just need the corresponding IV offset of the block in order to get the plaintext of a specific block.

1. **Obtain the files Alice. bmp and Alice. jpg from the textbook website.**
2. Use the TEA cipher to encrypt Alice. bmp in ECB mode, leaving the first 10 blocks unencrypted. View the encrypted image. What do you see? Explain the result.

I can clearly see the content of the picture, although the colors have been altered. The white spaces of the original image have been encrypted into repetitive lines of color, and the black lines of the original image have been encrypted into random repetitive colors. Although the image was encrypted, and we cannot see the original color of the image, we can still clearly see the content of the image. (Encrypted image included “EncryptedAlice.bmp”)

1. Use the TEA cipher to encrypt Alice. jpg in ECB mode, leaving the first 10 blocks unencrypted. View the encrypted image. What do you see? Explain the result.

The encryption of a jpg image is no longer a jpg image, when trying to open the file, it is shown as corrupt and nothing can be seen. I believe this is happening because in bmp file, the file properties are within the first 80 bits of data, but in jpg the file properties are not, and by encrypting it, we changed the file properties and it is no longer recognized as jpg.

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

Using repeated IV will lower the security levels to ECB mode. Because if the same IV is used with the same plaintext, then it will produce the same ciphertext, similar to ECB’s security issue.

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

Using the same IV for CTR will seriously undermine its security. Because CTR is encrypted in the way of C0 = P0 ⊕ F(IV ⊕ K), if we use the same IV every time, then F(IV ⊕ K) will be the same, therefore, let’s say we have the ciphertexts C1 = P1 ⊕ F(IV ⊕ K) and C2 = P2 ⊕ F(IV ⊕ K), then C1 ⊕ C2 = P1 ⊕ P2 and then Trudy can deduce the plaintext from there on. This example shows that if we use the same IV for CTR, the attacker doesn’t even need to know IV and he can easily get the plaintext.

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

The CBC mode will be more secure than the CTR mode, because in the CBC mode, using the same IV might cause the same blocks at the front, but it will cause less harm than directly letting Trudy get part of the plaintext.

1. **Suppose Alice has four blocks of plaintext, P0 , P1, P2, P3. She computes a MAC using key K1, and then CBC encrypts the data using key K2 to obtain C0 , C1, C2, C3 . Alice sends the IV, the ciphertext, and the MAC to Bob. Trudy intercepts the message and replaces C1 with X so that Bob receives IV, C0, X , C2, C3, and the MAC. Bob attempts to verify the integrity of the data by decrypting (using key K2) and then computing a MAC (using key K1) on the putative plaintext.**
   1. Show that Bob will detect Trudy’s tampering.

Bob will detect Trudy’s tampering because by changing C1 to X, when decrypting it will sub sequentially affect the decryption of C2 and since C2 is affected, C3 will also be affected. And when Bob tries to compute the MAC using K1 it will result in different MAC than the one originally sent by Alice.

REFERENCES

TEA cipher: <http://www.winterwell.com/software/TEA.php>

CTR with same IV: [http://crypto.stackexchange.com/questions/2991/why-must-iv-key-pairs-not-be-reused-in-ctr-mode](http://www.google.com/url?q=http%3A%2F%2Fcrypto.stackexchange.com%2Fquestions%2F2991%2Fwhy-must-iv-key-pairs-not-be-reused-in-ctr-mode&sa=D&sntz=1&usg=AFQjCNEJ7p6wH43SyhW5CMuh0CFfptAPvQ)

Classmates: Jay Patel, Jeffrey Su