FIT3031-Tutorial 2 Sample Solution

SYMMETRIC ENCRYPTION

Q1. What are the essential ingredients of a symmetric cipher?

Ans: Plaintext, encryption algorithm, secret key, ciphertext, decryption algorithm.

Q2. What are the two basic functions used in encryption algorithms

Ans: Permutation and substitution.

Q3. How many keys are required for two people to communicate via a symmetric cipher? How many keys are required for n people to communicate with each other securely?

Ans: One secret key. n(n-1)/2 keys

Q4. What is the difference between a block cipher and a stream cipher?

a. Why is it not desirable to reuse a stream cipher key?

b. Why do some block cipher modes of operation only use encryption while others use both encryption and decryption?

Ans: A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

a. If two plaintexts are encrypted with the same key using a stream cipher, then cryptanalysis is often quite simple. If the two ciphertext streams are XORed together, the result is the XOR of the original plaintexts. If the plaintexts are text strings, credit card numbers, or other byte streams with known properties, then cryptanalysis may be successful.
b. In some modes, the plaintext does not pass through the encryption function, but is XORed with the output of the encryption function. The math works out that for decryption in these cases, the encryption function must also be used.

Q5. What are the two general approaches to attacking a cipher?

Ans: Cryptanalysis and brute force.

Q6. List and briefly define types of cryptanalytic attacks based on what is known to the attacker.

Ans:

Type of Attack	Known to Cryptanalyst
Ciphertext only	•Encryption algorithm •Ciphertext to be decoded
Known plaintext	•Encryption algorithm •Ciphertext to be decoded •One or more plaintext-ciphertext pairs formed with the secret key
Chosen plaintext	•Encryption algorithm •Ciphertext to be decoded •Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	•Encryption algorithm •Ciphertext to be decoded •Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	•Encryption algorithm •Ciphertext to be decoded •Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key •Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Table 2.1 Types of Attacks on Encrypted Messages

Q7. What is triple encryption? Why is the middle portion of 3DES a decryption rather than an encryption?

Ans: With triple encryption, a plaintext block is encrypted by passing it through an encryption algorithm; the result is then passed through the same encryption algorithm again; the result of the second encryption is passed through the same encryption algorithm a third time. Typically, the second stage uses the decryption algorithm rather than the encryption algorithm.
There is no cryptographic significance to the use of decryption for the second stage. Its

only advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single DES by repeating the key. Hence the middle portion of 3DES is a decryption rather than an encryption

Q8. List ways in which secret keys can be distributed to two communicating parties.

Ans: For two parties A and B, key distribution can be achieved in a number of ways, as follows:

1. A can select a key and physically deliver it to B.
2. A third party can select the key and physically deliver it to A and B.
3. If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
4. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B.

Q9. What is the difference between a session key and a master key?

Ans: A session key is a temporary encryption key used between two principals. A master key is a long-lasting key that is used between a key distribution center and a principal for the purpose of encoding the transmission of session keys. Typically, the master keys are distributed by non-cryptographic means.

Q10. What is a key distribution center?

Ans: A key distribution center is a system that is authorized to transmit temporary session keys to principals. Each session key is transmitted in encrypted form, using a master key that the key distribution center shares with the target principal.

Problems

1. Prove the following:
   a. $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
   b. $A \oplus A = 0$
   c. $A \oplus 0 = A$
   d. $A \oplus 1$ = bitwise complement of A = A'
   e. $(A \oplus B)' = A' \oplus B = A \oplus B'$
   f. $A' \oplus B' = A \oplus B$

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

Ans. Let us perform the operations on 1 bit numbers and it is true for n bit strings.

a. $(A \oplus B) \oplus C = A \oplus (B \oplus C)$

A	**B**	**C**	**$(A \oplus B) \oplus C$**	**$A \oplus (B \oplus C)$**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

b. $A \oplus A = 0$

A	**A**	**$A \oplus A = 0$**
0	0	0
1	1	0

c. $A \oplus 0 = A$

A		**$A \oplus 0 = A$**
0	0	0
1	0	1

d. $A \oplus 1 = A'$

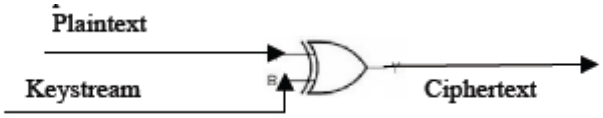
A		**$A \oplus 1 = A'$**
0	1	1
1	1	0

e. The equality can be shown by listing all 1-bit possibilities:

A	B	$A \oplus B$	$(A \oplus B)'$	$A' \oplus B$
0	0	0	1	1
0	1	1	0	0
1	0	1	0	0
1	1	0	1	1

f. We also need the equality A ⊕ B = A' ⊕ B', which is easily seen to be true.

2. Stream Cipher:



What is the value of ciphertext if:
Plaintext : 1010101010100
Keystream : 1100110001001

Ans :Ciphertext = Plaintext XOR keystream = 0110011011101

3. With the ECB mode, if there is an error in a block of the transmitted ciphertext, only the corresponding plaintext block is affected. However, in the CBC mode this error propagates. For example, an error in the transmitted C_1 (Figure 6.4 below) obviously corrupts P_1.
   a. Are any blocks beyond P_2 affected?
   b. Suppose there is a bit error in the source version of P_1. Through how many ciphertext blocks is this error propagated? What is the effect at the receiver?
   c. Is it possible to perform encrytions operation in parallel on multple blocks of plaintext in the CBC mode? How about decryption?
   d. Suppose there is an error in a block of ciphertext on transmission using CBC, with reference to the Figure given below. What effect is produced on the recovered plaintext blocks?

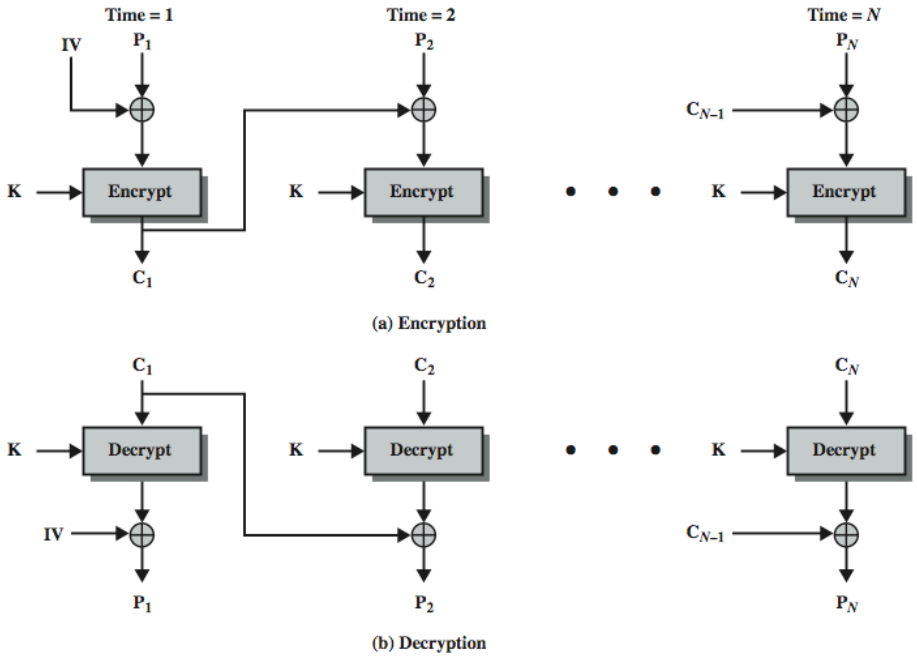


Figure 6.4 Cipher Block Chaining (CBC) Mode

Ans:

a. No. For example, suppose C1 is corrupted. The output block P3 depends only on the input blocks C2 and C3.

b. An error in P1 affects C1. But since C1 is input to the calculation of C2, C2 is affected. This effect carries through indefinitely, so that all ciphertext blocks are affected. However, at the receiving end, the decryption algorithm restores the correct plaintext for blocks except the one in error. You can show this by writing out the equations for the decryption. Therefore, the error only effects the corresponding decrypted plaintext block

c. In CBC encryption, the input block to each forward cipher operation (except the first) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel. In CBC decryption, however, the input blocks for the inverse cipher function (i.e., the ciphertext blocks) are immediately available, so that multiple inverse cipher operations can

be performed in parallel.

d. If an error occurs in transmission of ciphertext block Ci, then this error propagates to the recovered plaintext blocks Pi and Pi+1.

4. For each of the modes ECB, CBC, CTR shown in Figures 6.3, 6.4 and 6.7 respectively:

a. Identify which decrypted plaintext blocks P_x will be corrupted if there is an error in block C_4 of the transmitted ciphertext.

b. Assuming that the ciphertext contains N blocks, and there was a bit error in the source version of P_3, identify through how many ciphertext blocks this error is propagated.

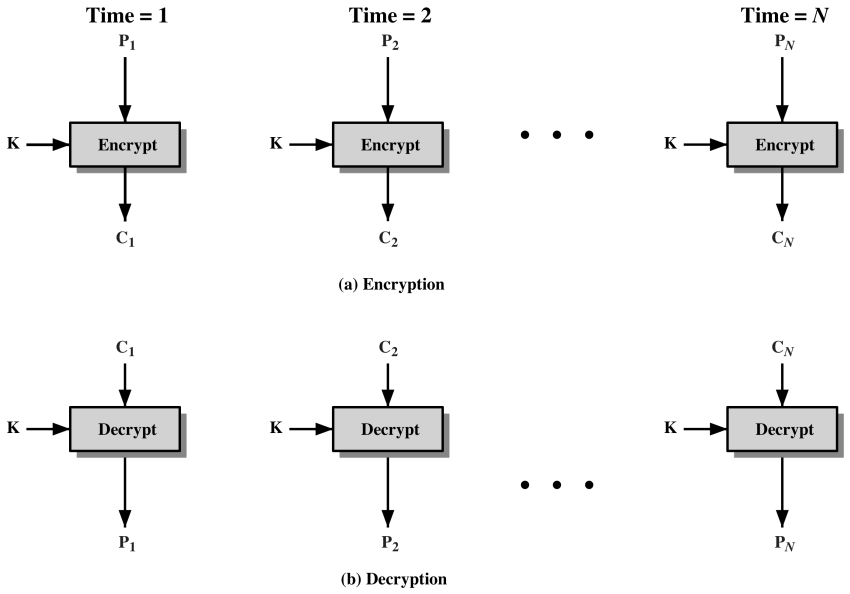


Figure 6.3 Electronic Codebook (ECB) Mode

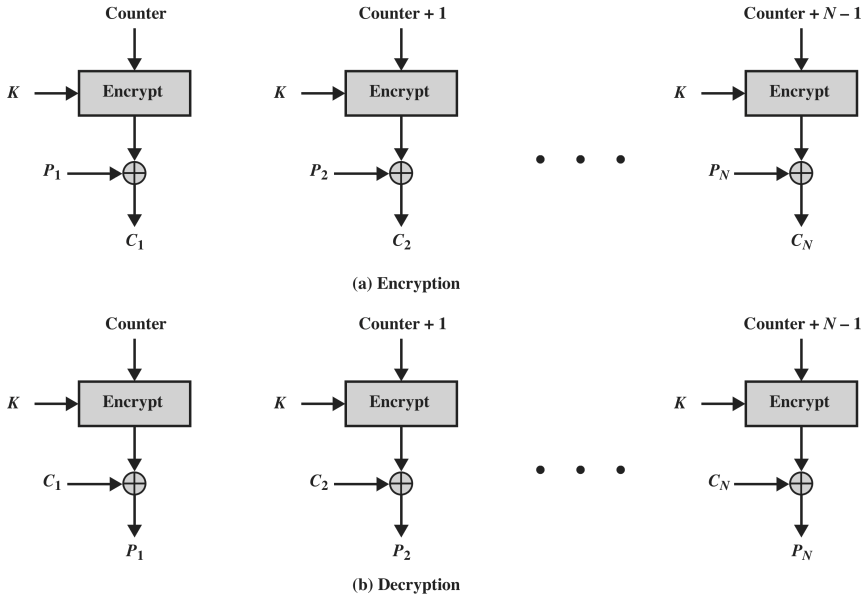


Figure 6.7 Counter (CTR) Mode

Ans: The question assumes that there was an error in block C_4 of the transmitted ciphertext.
ECB mode: In this mode, ciphertext block C_i is used only as input for the direct dencryption of plaintext block P_i. Therefore, a transmission error in block C_4 will only corrupt block P_4 of the decrypted plaintext.

CBC mode: In this mode, ciphertext block C_i is used as input to the XOR function when obtaining plaintext blocks P_i and P_{i+1}. Therefore, a transmission error in block C_4 will corrupt blocks P_4 and P_5 of the decrypted plaintext, but will not propagate to any of the other blocks.

CTR mode: In this mode, ciphertext block C_i, as well as the encrypted counter t_i, are used only as input for the direct decryption of plaintext block P_i. Therefore, a transmission error in block C_4 will only corrupt block P_4 of the decrypted plaintext.

5. Is it possible to perform encryption operations in parallel on multiple blocks of plaintext in OFB mode which is shown in Figure 6.6? How about decryption?

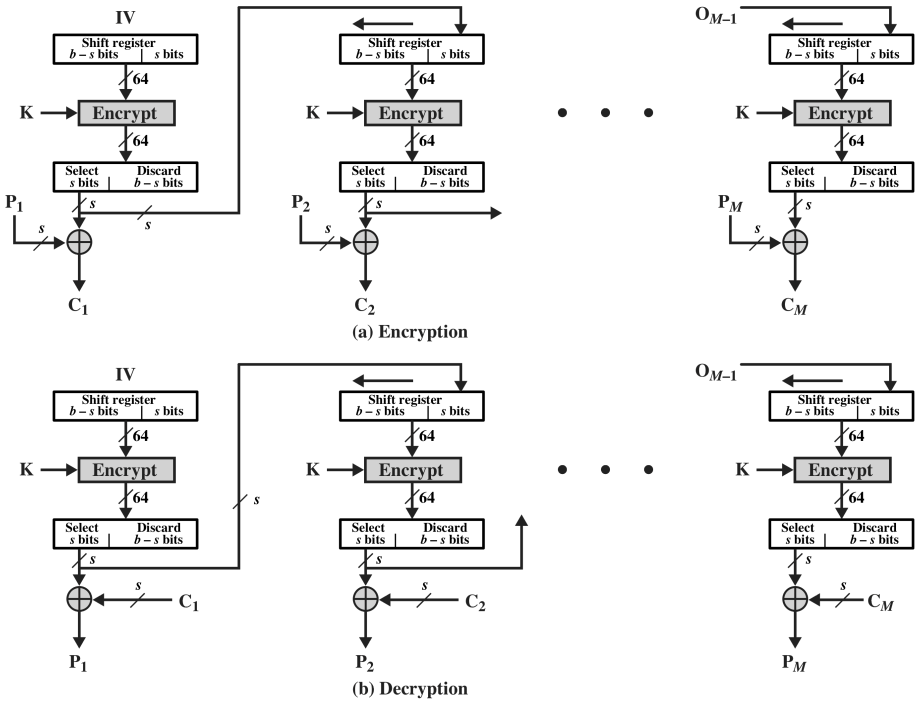


Figure 6.6 *s*-bit Output Feedback (OFB) Mode

Ans: Parallel operations are possible for both encryption and decryption in OFB mode. To see this, note that the chaining taking place in OFB is only of the IV being re-encrypted over and over again, each re-encryption being XORed with a block of plaintext. Thus, if the IV has been encrypted *h* times, *h* blocks of ciphertext can then be processed in parallel. The same holds for decryption.

6. Consider a block cipher algorithm with the following properties:

- Input and output block length of 64 bits and the key size is 56 bits
- Given a key K, the key scheduling requires 2 microseconds (2 x 10^{-6} secs)
- After the key scheduling produces all the sub-keys (if required), the encryption of a single block of 64 bits block takes 0.5 microseconds.

Compute the following information:

a. The total time required (of course in microseconds) to encrypt **1MBytes** (**2^{20}** bytes) of data.

Ans: First we need to find the number of 64-bit blocks in 1MByte of data as

# of bits in 1MB = 2^{20} bytes * 8 bits/byte

= 8,388,608 =2^{23} bits

Number of data blocks **= 8,388,608 / 64 bits**

= 131,072 blocks. = 2^{17} **blocks of 64 bits each**

It is now simply a matter of recognizing that the key K will be scheduled only once for this encryption, and that we need to encrypt 131,072 blocks of data.

Time= **2 microseconds + 2^{17} * 0.5 microseconds = 65,536 + 2 = 65,538 microseconds**

b. Given 2 values C and M such that $C = E_K(M)$ under the unknown key value K, how many years (at most) are required to crack the cipher on a single computer.

Ans : The second part seeks the amount of time, at most, it would take to crack the cipher given ciphertext C and the related plaintext M. In order to do this, it is necessary to search the entire key space. Because a key is 56 bits long, the key space is then

$$2^{56} = 72,057,594,037,927,936$$

Now we know how many keys we need to try before we find the right one, we must recognize that we only need to test a single block of data. Then each trial requires key scheduling plus the time to encrypt/decrypt (depending on which one you choose). Thus, the equation becomes

(2^{56} (2 microseconds + 0.5 microseconds)) * 0.000001 secs ~= 1.8x 10^{11} secs

Translate this value into years **~= 2,084,999** days **~=5712** years!!