# FIT2093: Sample Tutorial 5 Solutions

## Symmetric Key Cryptography

## Review

1. What is the purpose of the S-boxes in DES?
   The S-box is a substitution function that introduces nonlinearity and adds to the complexity of the transformation.

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

3. What is triple encryption?
   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.

4. How many keys are used in triple encryption?
   Triple encryption can be used with three distinct keys for the three stages; alternatively, the same key can be used for the first and third stage.

5. Does a substitution need to be a permutation of the plaintext symbols? Why or why not?
   No. A substitution can be to an entirely different alphabet. One plaintext symbol can convert to several ciphertext symbols, or vice versa. For example, Morse code is a form of substitution of alphabetic letters to dots and dashes. Two plaintext characters could map the same ciphertext character as long as the recipient could distinguish between the two.

6. Explain why the product of two relatively simple ciphers, such as a substitution and a transposition, can achieve a high degree of security.
   Each cipher contributes its own strength, so ideally the strength of the product is at least the product of the strengths of the input ciphers. A substitution cipher contributes confusion, whereas a transposition performs diffusion. The DES and AES algorithms both use a combination of relatively simple functions. Obviously, however, just composing two ciphers is not guaranteed to result in a stronger combination.

7. What is a meet-in the-middle attack?
   This is an attack used against a double encryption algorithm and requires a known (plaintext, ciphertext) pair. In essence, the plaintext, P, is encrypted to produce an intermediate value in the double encryption, and the ciphertext, C, is decrypted to produce an intermediate value in the double encryption.
   - works whenever use a cipher twice
   - since $X = EK_1(P) = DK_2(C)$
   - attack by encrypting P with all keys and store
   - then decrypt C with keys and match X value
   - Check these keys with the another pair of (P,C), if it is valid, you have the guessed the keys $K_1$ and $K_2$.
   - can show that the effort is of the order of $O(2^{56})$ steps instead of $O(2^{112})$ as was expected.

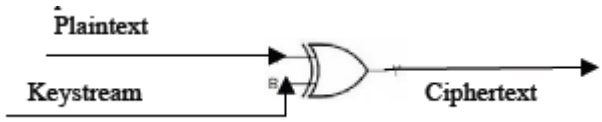
   Table lookup techniques can be used in such a way to dramatically improve on a brute-force try of all pairs of keys.

## Problems

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

   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.

2. Stream Cipher:




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

   Ciphertext = Plaintext XOR keystream = 0110011011101

3. Perform encryption and decryption using TPC for Plaintext “25” and key [1,6]

   Plaintext as blocks: 0000001000000101
   Key [1, 6]: 0000000100000110
   Encryption:

– Step 1: number of 0s before a non-zero digit in the key is 7, hence we left shift the bits in the message by 7 positions.
> Scrambled message: 0000001010000001
– Step 2: perform XOR operation on the scrambled message and the key.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSG | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Key | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Decryption:
– Step 1: perform XOR between the ciphertext and the key.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Key | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| MSG | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

– Step 2: Shift the result of the XOR operation 7 bits to the right.
> 0000001000000101

4. The role of the S-boxes in the function F of DES is illustrated in Figure 1 and 2. The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output. These transformations are defined in Table 1, which is interpreted as follows: The first and last bits of the input to box $S_i$ form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for $S_i$. The middle four bits select one of the sixteen columns. Find the output for the following input:

   a. $S_1$ (011001)
   $S_1$ for input 011001, the row is 01 (row 1) and the column is 1100 (column 12). The value in row 1, column 12 is 9, so the output is 1001.

   b. $S_5$ (000000)

   $S_5$ for input 000000, the row is 00 (row 0) and the column is 0000 (column 0). The value in row 0, column 0 is 2, so the output is 0010.

   c. $S_6$ (111111)

   $S_6$ for input 111111, the row is 11 (row 3) and the column is 1111 (column 15). The value in row 3, column 15 is 13, so the output is 1101.

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

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.

6. 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.

      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.

   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.

      The question assumes that the ciphertext contains N blocks, and that there was a bit error in the source version of $P_3$.
      **ECB mode:** In this mode, ciphertext block $C_i$ is generated by direct encryption of plaintext block $P_i$, independent of the other plaintext or ciphertext blocks. Therefore, a bit error in block $P_3$ will only affect ciphertext block $C_3$ and will not propagate further. Thus, only one ciphertext block will be corrupted.

      **CBC mode:** In this mode, ciphertext block $C_i$ is generated by XORing plaintext block $P_i$ with ciphertext block $C_{i-1}$. Therefore, a bit error in block $P_3$ will affect ciphertext block $C_3$, which in turn will affect ciphertext block $C_4$ and so forth, and therefore the error will propagate through all remaining ciphertext blocks. Thus, N-2 ciphertext block will be corrupted.

**CTR mode:** In this mode, ciphertext block $C_i$ is generated by applying the XOR function to plaintext block $P_i$ and the encrypted counter $t_i$, independent of the other plaintext or ciphertext blocks. Therefore, a bit error in block $P_3$ will only affect ciphertext block $P_3$ and will not propagate further. Thus, only one ciphertext block will be corrupted.

7. 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?

   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.

8. If a bit error occurs in the transmission of a ciphertext charater in 8-bit CFB mode shown in Figure 6.5, how far does the error propagate?

   Nine plaintext characters are affected. The plaintext character corresponding to the ciphertext character is obviously altered. In addition, the altered ciphertext character enters the shift register and is not removed until the next eight characters are processed.

9. 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 ($2 \times 10^{-6}$ 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** ($\mathbf{2^{20}}$ bytes) of data.

      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.

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!!

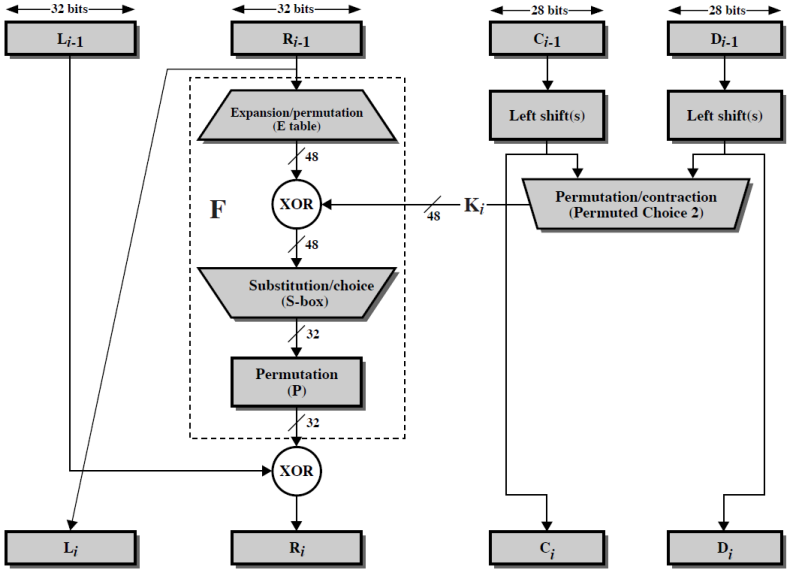



**Figure 1 Single Round of DES Algorithm**

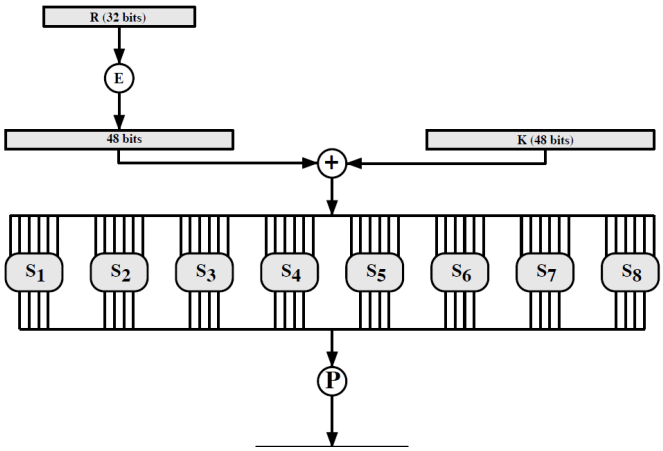



**Figure 2 Calculation of F(R, K)**

**Table 1 Definition of DES S-Boxes**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |
| $S_2$ | 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| | 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| | 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| | 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |
| $S_3$ | 10 | 0 | 9 | 14 | 6 | 3 | 15 | 5 | 1 | 13 | 12 | 7 | 11 | 4 | 2 | 8 |
| | 13 | 7 | 0 | 9 | 3 | 4 | 6 | 10 | 2 | 8 | 5 | 14 | 12 | 11 | 15 | 1 |
| | 13 | 6 | 4 | 9 | 8 | 15 | 3 | 0 | 11 | 1 | 2 | 12 | 5 | 10 | 14 | 7 |
| | 1 | 10 | 13 | 0 | 6 | 9 | 8 | 7 | 4 | 15 | 14 | 3 | 11 | 5 | 2 | 12 |
| $S_4$ | 7 | 13 | 14 | 3 | 0 | 6 | 9 | 10 | 1 | 2 | 8 | 5 | 11 | 12 | 4 | 15 |
| | 13 | 8 | 11 | 5 | 6 | 15 | 0 | 3 | 4 | 7 | 2 | 12 | 1 | 10 | 14 | 9 |
| | 10 | 6 | 9 | 0 | 12 | 11 | 7 | 13 | 15 | 1 | 3 | 14 | 5 | 2 | 8 | 4 |
| | 3 | 15 | 0 | 6 | 10 | 1 | 13 | 8 | 9 | 4 | 5 | 11 | 12 | 7 | 2 | 14 |
| $S_5$ | 2 | 12 | 4 | 1 | 7 | 10 | 11 | 6 | 8 | 5 | 3 | 15 | 13 | 0 | 14 | 9 |
| | 14 | 11 | 2 | 12 | 4 | 7 | 13 | 1 | 5 | 0 | 15 | 10 | 3 | 9 | 8 | 6 |
| | 4 | 2 | 1 | 11 | 10 | 13 | 7 | 8 | 15 | 9 | 12 | 5 | 6 | 3 | 0 | 14 |
| | 11 | 8 | 12 | 7 | 1 | 14 | 2 | 13 | 6 | 15 | 0 | 9 | 10 | 4 | 5 | 3 |
| $S_6$ | 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 11 |
| | 10 | 15 | 4 | 2 | 7 | 12 | 9 | 5 | 6 | 1 | 13 | 14 | 0 | 11 | 3 | 8 |
| | 9 | 14 | 15 | 5 | 2 | 8 | 12 | 3 | 7 | 0 | 4 | 10 | 1 | 13 | 11 | 6 |
| | 4 | 3 | 2 | 12 | 9 | 5 | 15 | 10 | 11 | 14 | 1 | 7 | 6 | 0 | 8 | 13 |
| $S_7$ | 4 | 11 | 2 | 14 | 15 | 0 | 8 | 13 | 3 | 12 | 9 | 7 | 5 | 10 | 6 | 1 |
| | 13 | 0 | 11 | 7 | 4 | 9 | 1 | 10 | 14 | 3 | 5 | 12 | 2 | 15 | 8 | 6 |
| | 1 | 4 | 11 | 13 | 12 | 3 | 7 | 14 | 10 | 15 | 6 | 8 | 0 | 5 | 9 | 2 |
| | 6 | 11 | 13 | 8 | 1 | 4 | 10 | 7 | 9 | 5 | 0 | 15 | 14 | 2 | 3 | 12 |
| $S_8$ | 13 | 2 | 8 | 4 | 6 | 15 | 11 | 1 | 10 | 9 | 3 | 14 | 5 | 0 | 12 | 7 |
| | 1 | 15 | 13 | 8 | 10 | 3 | 7 | 4 | 12 | 5 | 6 | 11 | 0 | 14 | 9 | 2 |
| | 7 | 11 | 4 | 1 | 9 | 12 | 14 | 2 | 0 | 6 | 10 | 13 | 15 | 3 | 5 | 8 |

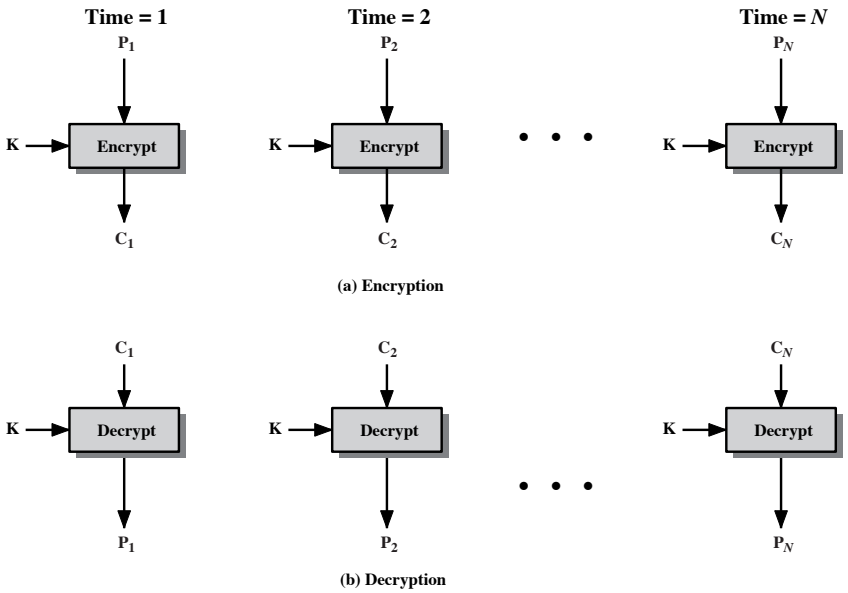



**Figure 6.3 Electronic Codebook (ECB) Mode**

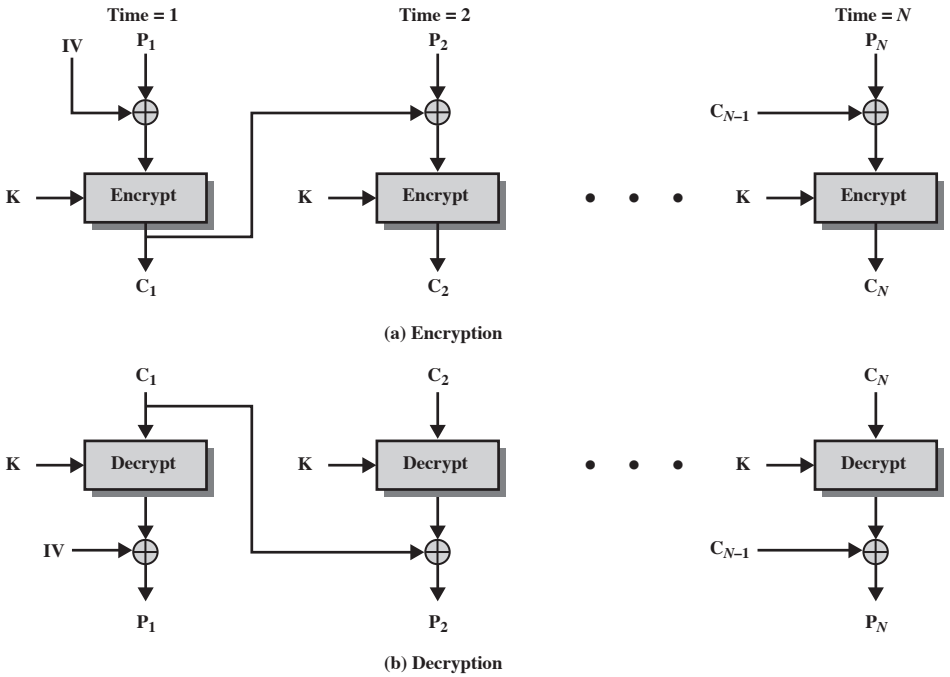



**Figure 6.4 Cipher Block Chaining (CBC) Mode**

---

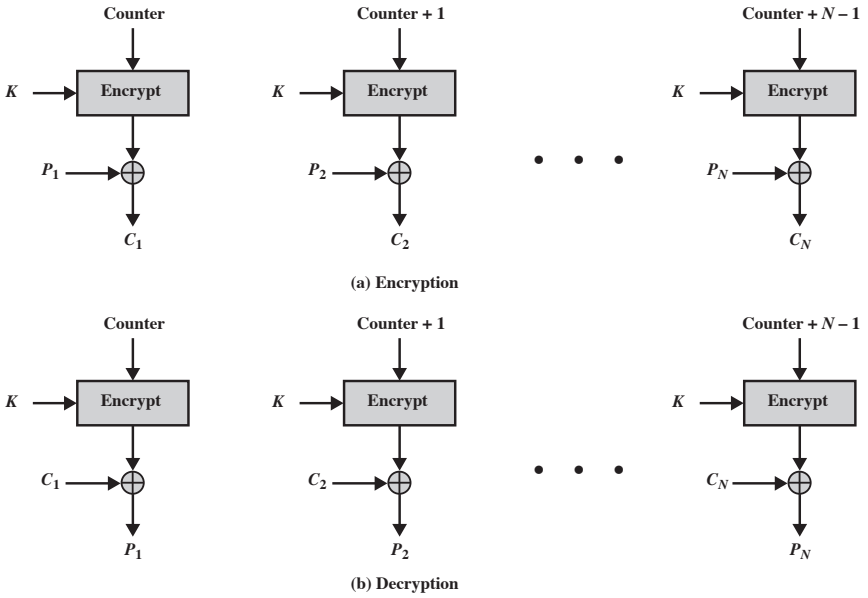



**Figure 6.7 Counter (CTR) Mode**

---

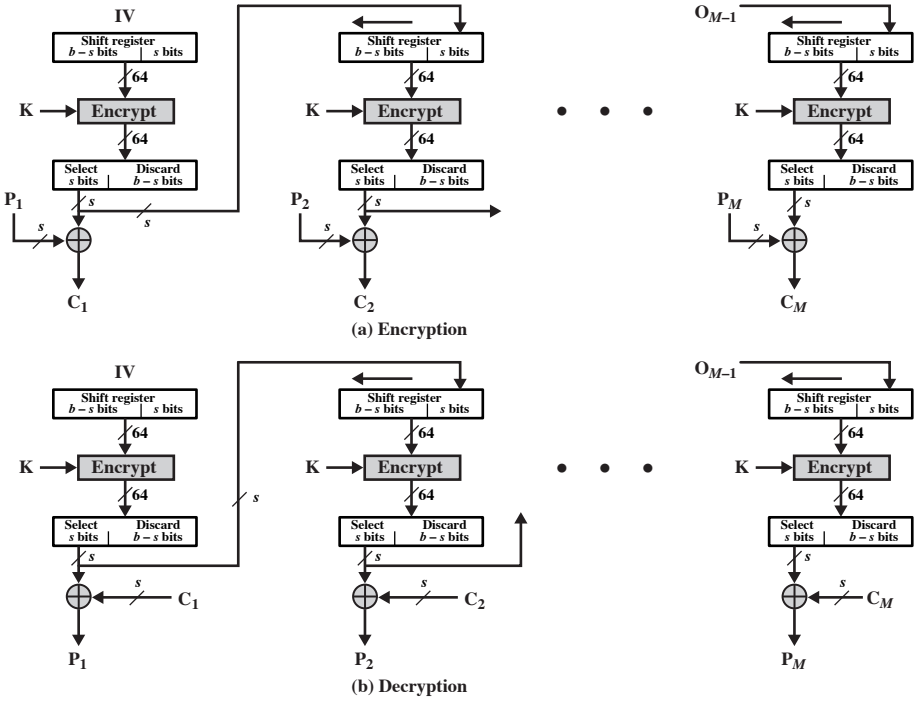



**Figure 6.6** *s*-bit Output Feedback (OFB) Mode

---

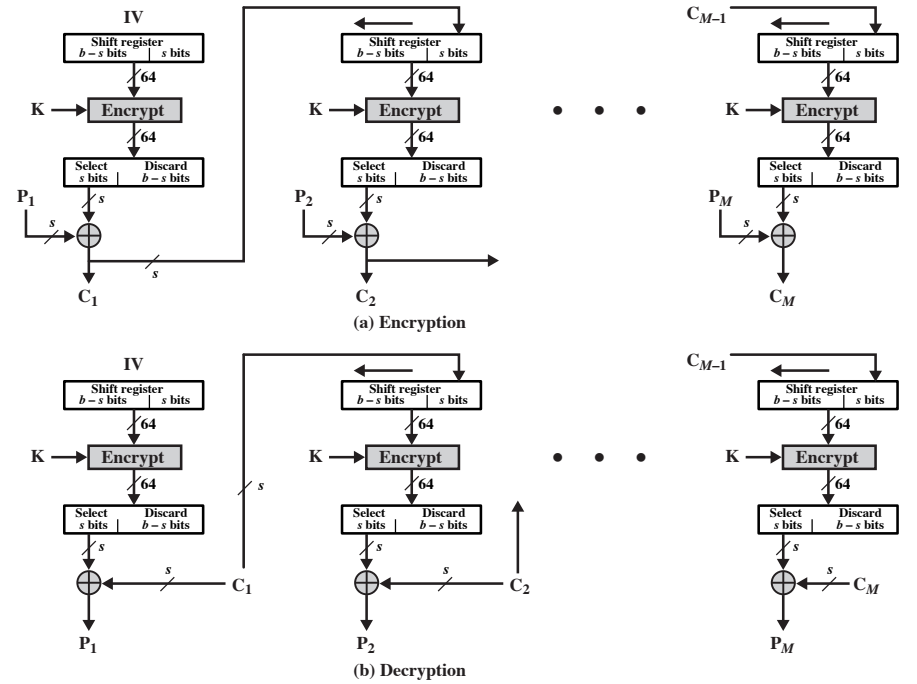



**Figure 6.5** *s*-bit Cipher Feedback (CFB) Mode