Block Cipher Modes of Operation

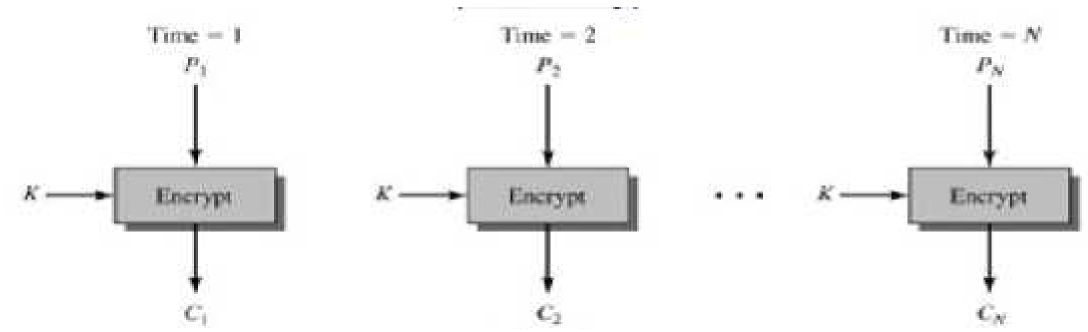
A block cipher algorithm is a basic building block for providing data security. In essence, a mode of operation is a technique for enhancing the effect of a cryptographic algorithm or adapting the algorithm for an application, such as applying a block cipher to a sequence of data blocks or a data stream.

There are five modes of operation standardization. These modes are intended for use with any symmetric block cipher, including triple DES and AES. These modes are:

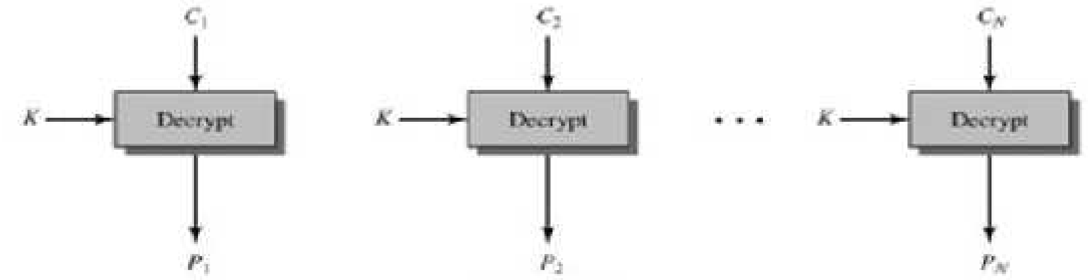
1- Electronic Codebook (ECB).
2- Cipher Block Chaining (CBC).
3- Cipher Feedback (CFB).
4- Output Feedback (OFB).
5- Counter (CTR).

1- Electronic Codebook Mode

The simplest mode is the electronic codebook (ECB) mode, in which plaintext is handled one block at a time and each block of plaintext is encrypted using the same key (Figure 9.1). The term *codebook* is used because, for a given key, there is a unique ciphertext for every *b*-bit block of plaintext. Therefore, we can imagine a gigantic codebook in which there is an entry for every possible *b*-bit plaintext pattern showing its corresponding ciphertext.



Encryption



Decryption

Figure (9.1) Electronic Codebook (ECB) Mode

The ECB method is ideal for a short amount of data, such as an encryption key. Thus, if you want to transmit a DES key securely, ECB is the appropriate mode to use.

The most significant characteristic of ECB is that the same *b*-bit block of plaintext, if it appears more than once in the message, always produces the same ciphertext. So, For lengthy messages, the ECB mode may not be secure.

2- Cipher Block Chaining Mode

To overcome the security deficiencies of ECB, we would like a technique in which the same plaintext block, if repeated, produces different ciphertext blocks. A simple way to satisfy this requirement is the cipher block chaining (CBC) mode (Figure 9.2).

In this scheme, the input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block; the same key is used for each block. In effect, we have chained together the processing of the sequence of plaintext blocks. The input to the encryption function for each plaintext block bears no fixed relationship to the plaintext block.

Therefore, repeating patterns of *b* bits are not exposed.

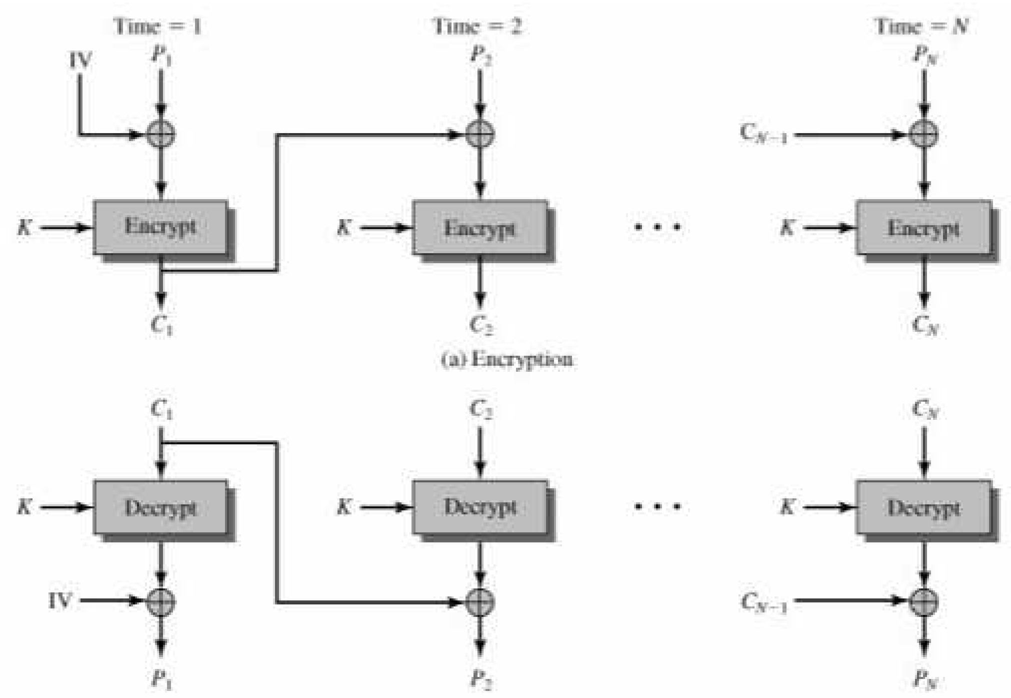


Figure (9.2) Cipher Block Chaining (CBC) Mode

For decryption, each cipher block is passed through the decryption algorithm. The result is XORed with the preceding ciphertext block to produce the plaintext block. To see that this works, we can write

$$C_j = E(K, [C_{j-1} \oplus P_j])$$

Then

$$D(K, C_j) = D(K, E(K, [C_{j-1} \oplus P_j]))$$

$$D(K, C_j) = C_{j-1} \oplus P_j$$

$$C_{j-1} \oplus D(K, C_j) = C_{j-1} \oplus C_{j-1} \oplus P_j = P_j$$

To produce the first block of ciphertext, an initialization vector (IV) is XORed with the first block of plaintext. On decryption, the IV is XORed with the output of the decryption algorithm to recover the first

block of plaintext. The IV is a data block that is that same size as the cipher block.

The IV must be known to both the sender and receiver but be unpredictable by a third party. For maximum security, the IV should be protected against unauthorized changes. This could be done by sending the IV using ECB encryption. One reason for protecting the IV is as follows: If an opponent is able to fool the receiver into using a different value for IV, then the opponent is able to invert selected bits in the first block of plaintext. To see this, consider the following:

$$C_1 = E(K, [IV \oplus P_1])$$

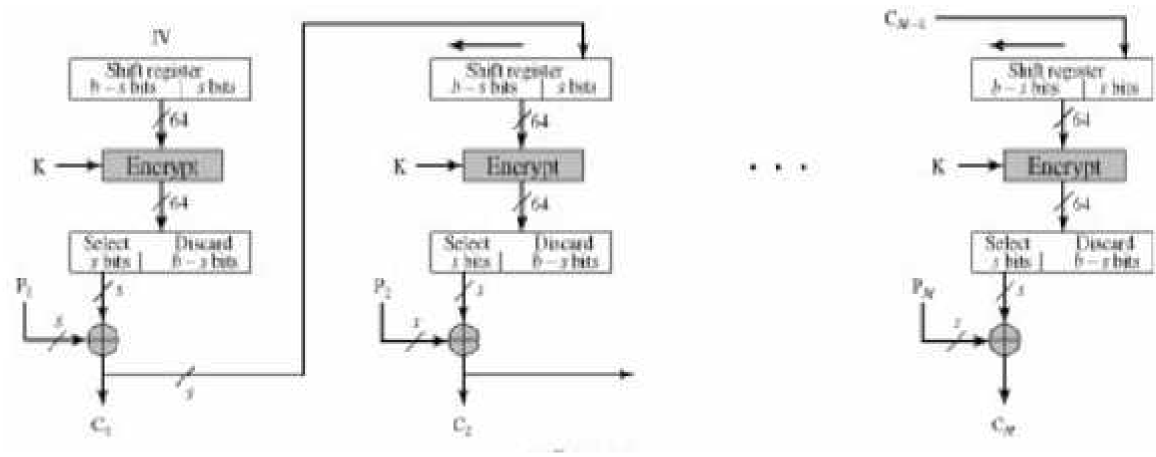
$$P_1 = IV \oplus D(K, C_1)$$

3- Cipher Feedback Mode(CFM):

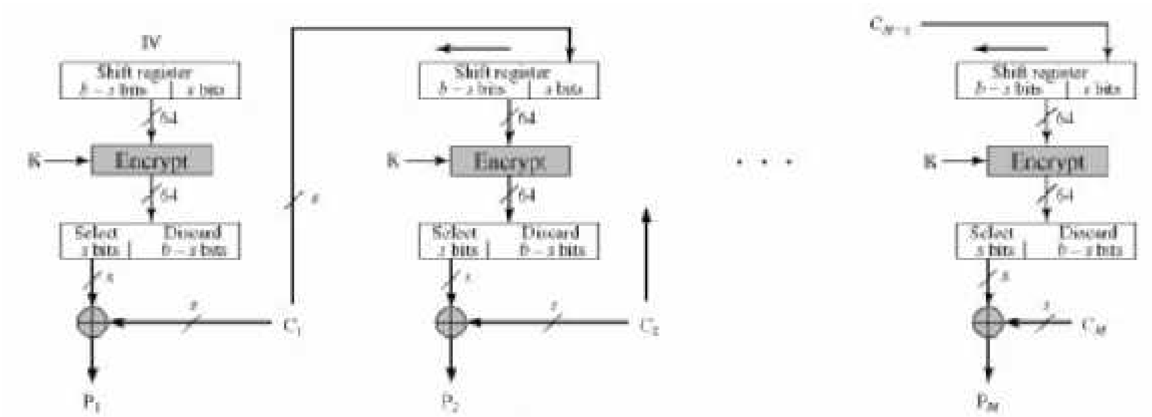
The DES scheme is essentially a block cipher technique that uses *b*-bit blocks. However, it is possible to convert DES into a stream cipher, using either the cipher feedback (CFB) or the output feedback mode.

A stream cipher eliminates the need to pad a message to be an integral number of blocks. It also can operate in real time. Thus, if a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character-oriented stream cipher.

One desirable property of a stream cipher is that the ciphertext be of the same length as the plaintext. Thus, if 8-bit characters are being transmitted, each character should be encrypted to produce a cipher text output of 8 bits. If more than 8 bits are produced, transmission capacity is wasted. Figure 9.3 depicts the CFB scheme.



a)Encryption



b)Decryption

Figure (9.3) The CFB scheme

In the figure, it is assumed that the unit of transmission is *s* bits; a common value is *s* = 8. As with CBC, the units of plaintext are chained together, so that the ciphertext of any plaintext unit is a function of all the preceding plaintext. In this case, rather than units of *b* bits, the plaintext is divided into *segments* of *s* bits.

First, consider encryption.

- The input to the encryption function is a *b*-bit shift register that is initially set to some initialization vector (IV).

- The leftmost (most significant) *s* bits of the output of the encryption function are XORed with the first segment of plaintext *P1* to produce the first unit of ciphertext *C1*, which is then transmitted.
- In addition, the contents of the shift register are shifted left by *s* bits and *C1* is placed in the rightmost (least significant) *s* bits of the shift register. This process continues until all plaintext units have been encrypted.

For decryption,

- The same scheme is used, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit.

Note that it is the *encryption* function (*X*) be defined as the most significant *s* bits of *X*. Then that is used, not the decryption function. This is easily explained. Let *S s*(*X*) be defined as the most significant *s* bits of *X*. Then

$$C_1 = P_1 \oplus S_s[E(K, IV)]$$

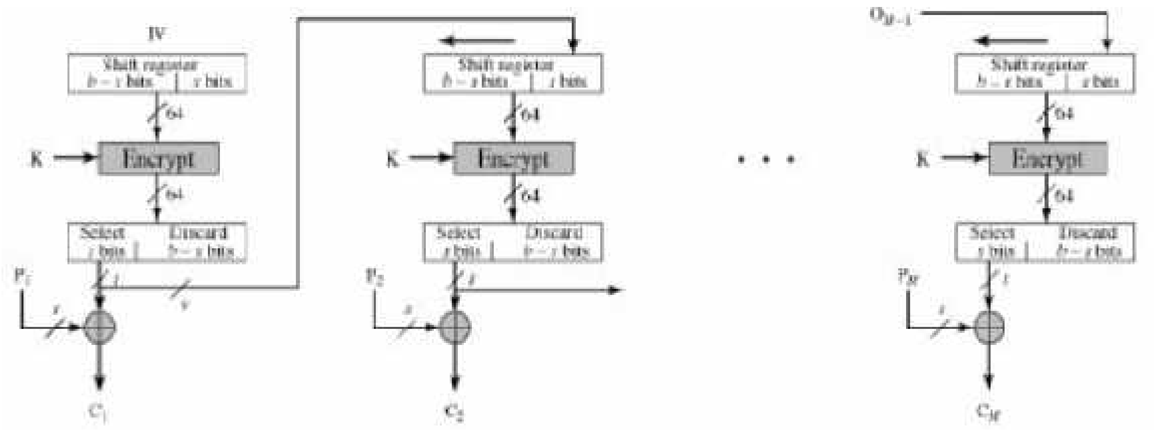
Therefore,

$$P_1 = C_1 \oplus S_s[E(K, IV)]$$

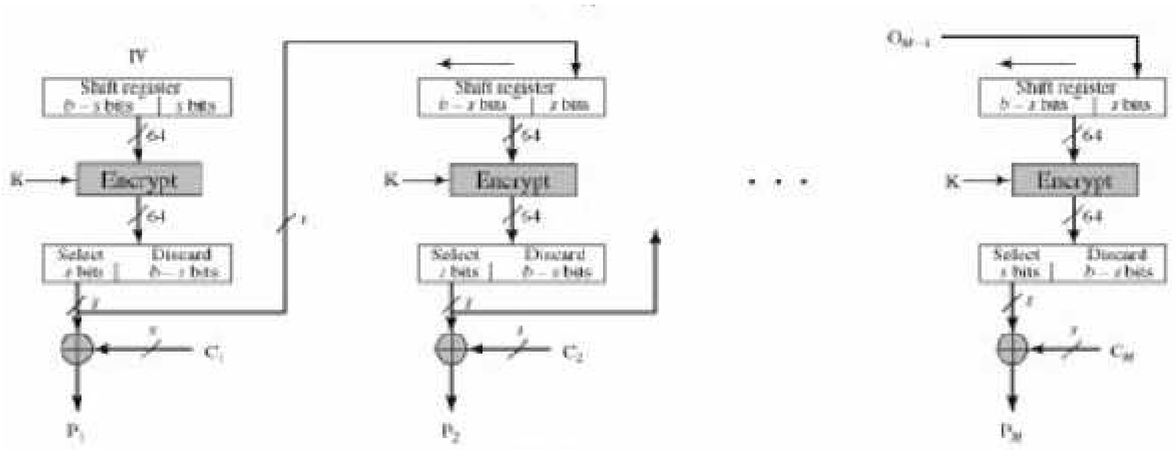
The same reasoning holds for subsequent steps in the process.

4- Output Feedback Mode

The output feedback (OFB) mode is similar in structure to that of CFB, as illustrated in Figure 9.4. As can be seen, it is the output of the encryption function that is fed back to the shift register in OFB, whereas in CFB the ciphertext unit is fed back to the shift register.



a) Encryption



b) Decryption

One advantage of the OFB method is that bit errors in transmission do not propagate. For example, if a bit error occurs in $C1$ only the recovered value of is $P1$ affected; subsequent plaintext units are not corrupted. With CFB, $C1$ also serves as input to the shift register and therefore causes additional corruption downstream.

The disadvantage of OFB is that it is more vulnerable to a message stream modification attack than is CFB. Consider that complementing a bit in the ciphertext complements the corresponding bit in the recovered plaintext. Thus, controlled changes to the recovered plaintext can be made. This may make it possible for an opponent, by making the necessary changes to the checksum portion of the message as well as to the data portion, to alter the ciphertext in such a way that it is not detected by an error- correcting code.

5- Counter Mode

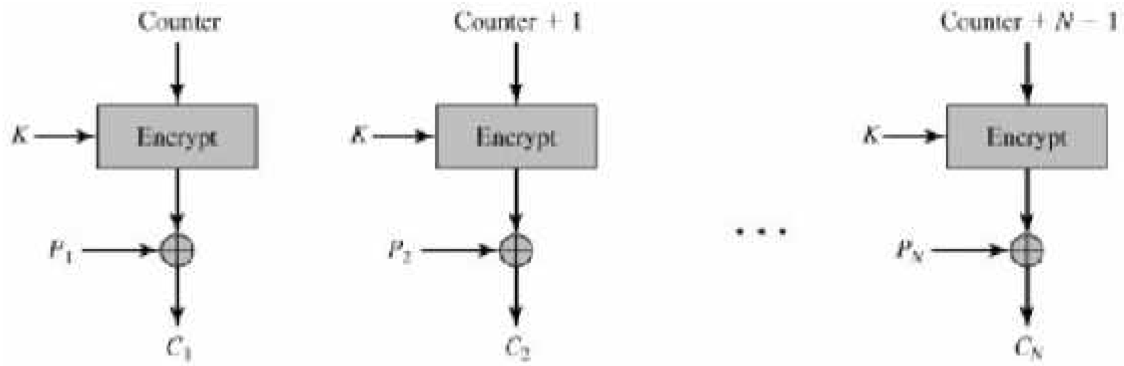
Figure 9.5 depicts the CTR mode. A counter, equal to the plaintext block size is used. that the counter value must be different for each plaintext block that is encrypted. Typically, the counter is initialized to some value and then incremented by 1 for each subsequent block (modulo 2*b* where *b* is the block size).

For encryption,

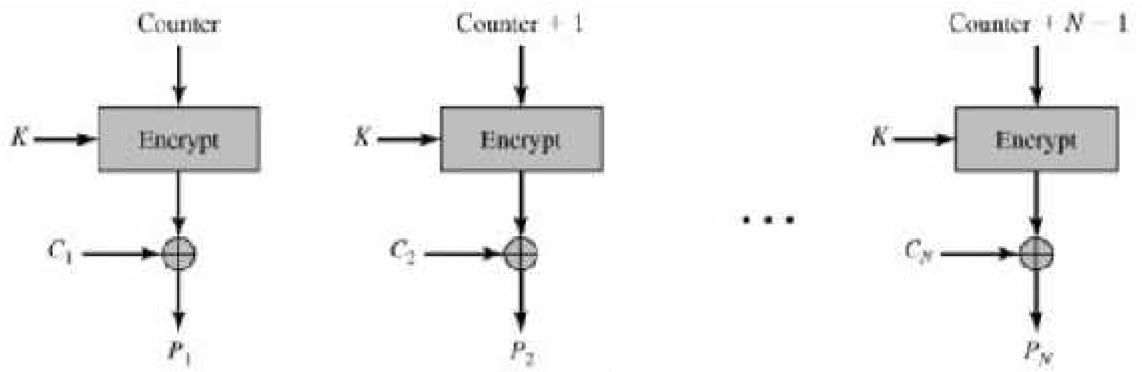
The counter is encrypted and then XORed with the plaintext block to produce the ciphertext block; there is no chaining.

For decryption,

The same sequence of counter values is used, with each encrypted counter XORed with a ciphertext block to recover the corresponding plaintext block.



a) Encryption



b) Decryption

Figure (9-5) CTR mode

The advantages of CTR mode:

- **Hardware efficiency:** Unlike the three chaining modes, encryption (or decryption) in CTR mode can be done in parallel on multiple blocks of plaintext or ciphertext. For the chaining modes, the algorithm must complete the computation on one block before beginning on the next block.
- **Software efficiency:** Similarly, because of the opportunities for parallel execution in CTR mode, processors that support parallel features, can be effectively used.
- **Preprocessing:** The execution of the underlying encryption algorithm does not depend on input of the plaintext or ciphertext. Therefore, if sufficient memory is available and security is maintained, preprocessing can be used to prepare the output of the encryption boxes that feed into the XOR functions in Figure 9.5. When the plaintext or ciphertext input is presented, then the only computation is a series of XORs.

Random access: The *i*th block of plaintext or ciphertext can be processed in random-access fashion. With the chaining modes, block *Ci* cannot be computed until the *i* - 1 prior block are computed. There may be applications in which a ciphertext is stored and it is desired to decrypt just one block; for such applications, the random access feature is attractive.

- **Simplicity:** Unlike ECB and CBC modes, CTR mode requires only the implementation of the encryption algorithm and not the decryption algorithm.