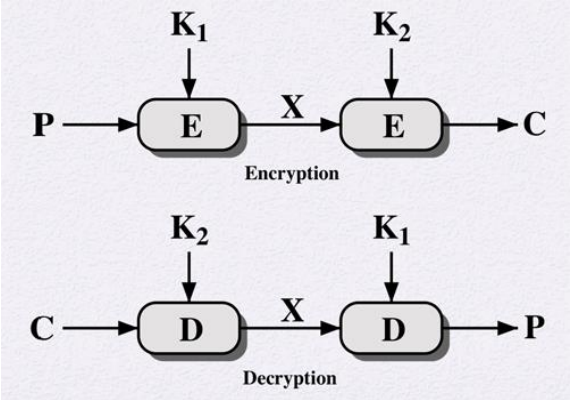
Multiple Encryption and Triple DES

- As we know that DES is venerable to brute-force attack, we are interested to find an alternative.
- One possible solution is to design completely new algorithm like AES.
- Second solution is to use DES multiple times.

Double DES

- The simplest form of multiple encryptions has two encryption stages and two keys and is known as Double DES.



- Given a plaintext P and two encryption keys K1 and K2, cipher text C is generated as: **C = E(K2, E(K1, P))**
- Decryption applies keys in reverse order: **P = D(K1, D(K2, C))**
- This scheme involves a key length of 56 * 2 = 112 bits, making Brute-Force attack impractical.
- However, other types of attacks are possible:

Reduction to a Single Stage

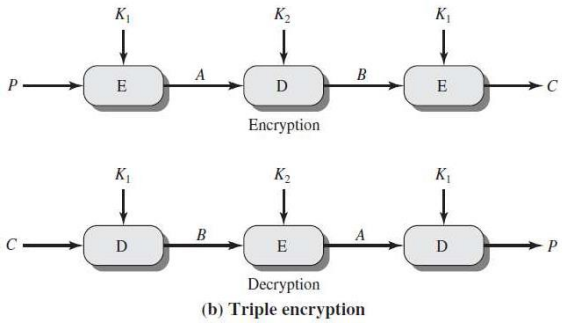
- If it is possible to find a key such that: **E(K2, E(K1, P)) = E(K3, P)** then double encryption, or any number of stages of multiple encryption with DES, would be useless.
- Because the result would be equivalent to a single encryption with a single 56- bit key.
- However, by the principle of reverse mapping, such a key is not possible.

Meet-In-The-Middle Attack

- This attack is based on the on the observation that if:
  $C = E(K_2, E(K_1, P))$, then
  $X = E(K_1, P) = D(K_2, C)$
- Given a known (P, C) pair, the attack proceeds as follows:
  - ✓ First, encrypt P for all 256 possible values of K1.
  - ✓ Store these results in a table and then sort the table by the values of X.
  - ✓ Decrypt C using all 256 possible values of K2.
  - ✓ Check the result against the table for a match after every decryption.
  - ✓ If a match occurs, then test the two resulting keys against a new known plaintext– ciphertext pair.
  - ✓ If the two keys produce the correct ciphertext, accept them as the correct keys.
  - ✓ For any given plaintext, 248 false alarms are possible since there are only 264 ciphertext values whereas 2112 key values.
  - ✓ Thus, the order of attack can be reduced to 248 instead of 2112.

Triple DES with Two Keys

- An alternative to the meet-in-the-middle attack is to use three stages of encryption with three or two different keys.



(b) Triple encryption

- The function follows an encrypt decrypt-encrypt (EDE) sequence.

  $C = E(K1, D(K2, E(K1, P)))$
  $P = D(K1, E(K2, D(K1, C)))$
- 3DES with two keys is a relatively popular alternative to DES.
- Currently, there are no practical cryptanalytic attacks on 3DES.
- Brute-force key search on 3DES is on the order of 2112 and the cost of differential cryptanalysis also has an exponential growth, compared to single DES.
- Several proposed attacks (though impractical) on 3DES are:

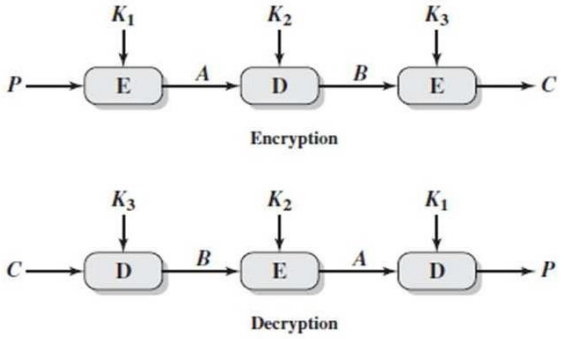
Chosen-Plaintext Attack

- Find plaintext values that gives A = 0.
- Then, use the meet-in-the- middle attack to determine the two keys.
- However, this attack requires 256 chosen plaintext-cipher text pairs which is impractical.

Known-Plaintext Attack

- This method does not require chosen plaintext-cipher text pairs but requires more effort.
- The attack is based on the observation that if an attacker knows A and C, then the problem reduces to that of an attack on double DES.
- The attack is as follows:
  - ✓ The attacker obtains n(P, C) pairs places them in Table 1 sorted on the values of P.
  - ✓ For an arbitrary value a for A, calculate the plaintext value that produces:
    Pi = D(i, a)
  - ✓ For each Pi that matches an entry in Table 1, create an entry in Table 2 that contains value of K1 and b that is obtained by decrypting the corresponding cipher text from Table 2.
    $B = D(i, C)$
  - ✓ Table 2 contains a number of candidate values of Ki. Now, for each of the 256 possible values of K2, calculate the second intermediate value for our chosen value of a:
    Bj = D(j, a)
  - ✓ At each step, look up Bj in Table 2. If there is a match, then the corresponding key i from Table 2 plus this value of j are candidate values for the unknown keys (K1, K2).
  - ✓ Test each candidate pair of keys on a few other plaintext–cipher text pairs. If a pair of keys produces the desired cipher text, the task is complete.
  - ✓ If no pair succeeds, repeat from step 1 with a new value of a.

Triple DES with Three Keys

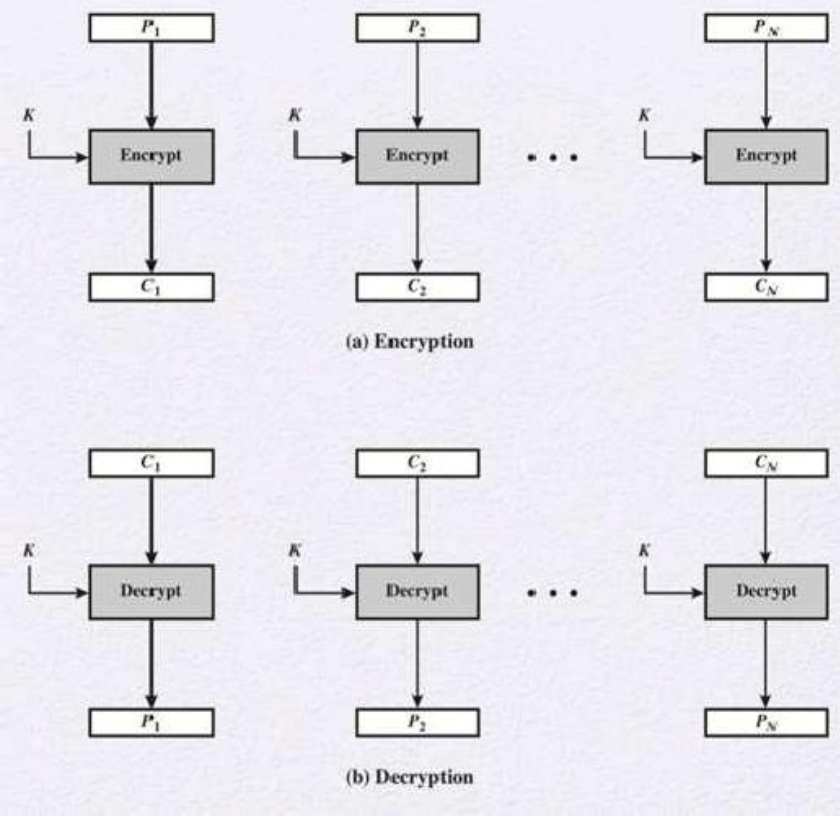


- Although the attacks just described appear impractical, anyone using two-key 3DES may feel some concern.
- In that case, three-key 3DES is the preferred alternative.
- Three-key 3DES has an effective key length of 168 bits and is defined as:
  C = E(K3, D(K2, E(K1, P)))
- Backward compatibility with DES is provided by putting K3 = K1 or K1 = K3.
- A number of Internet-based applications have adopted three-key 3DES, including PGP and S/MIME.

Modes of Operations

There are 5 modes of operation which are listed below.

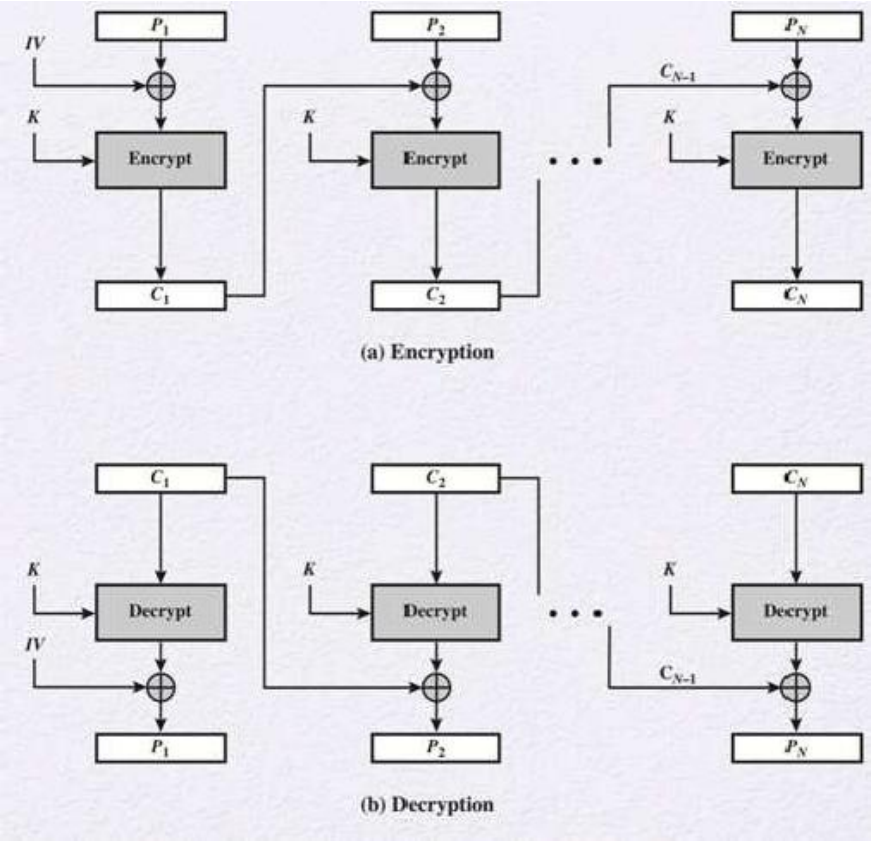
1. Electronic Codebook mode (ECB)



- This is the simplest mode in which plaintext is handled one block at a time and each block of plaintext is encrypted using the same key.

- The term codebook is used because, for a given key, there is a unique ciphertext for every -bit block of plaintext.
- Therefore, we can imagine a huge codebook in which there is an entry for every possible b-bit plaintext showing its corresponding ciphertext.
- For a message longer than b bits, the procedure is simply to break the message into b-bit blocks, padding the last block if necessary.
- Decryption is performed one block at a time, always using the same key.
- For lengthy messages, ECB mode may be not secure. If the message has repetitive elements, then these elements can be identified by the analyst.
- Thus, the ECB method is ideal for a short amount of data, such as an encryption key.

2. Cipher Block Chaining Mode (CBC)



- To overcome the security deficiencies of ECB, a technique is needed in which the same plaintext block, if repeated, produces different cipher text blocks.
- A simple way to satisfy this requirement is the cipher block chaining (CBC) which is shown in the figure.
- In this mode, the input to the encryption algorithm is the X-OR of the current plaintext block and the preceding ciphertext block; the same key is
- used for each block.
- The input to the encryption function for each plaintext block has no fixed relationship to the plaintext block.
- Therefore, repeating patterns will not produce same ciphertext.
- The last block is padded to a full b bits if it is a partial block.
- For decryption, each cipher block is passed through the decryption algorithm. The result is X-ORed with the preceding ciphertext block to produce the plaintext block.
- The expressions for CBC are:

Encryption:

$C\text{j} = E(K, [C\text{j} - 1 \oplus P\text{j})$

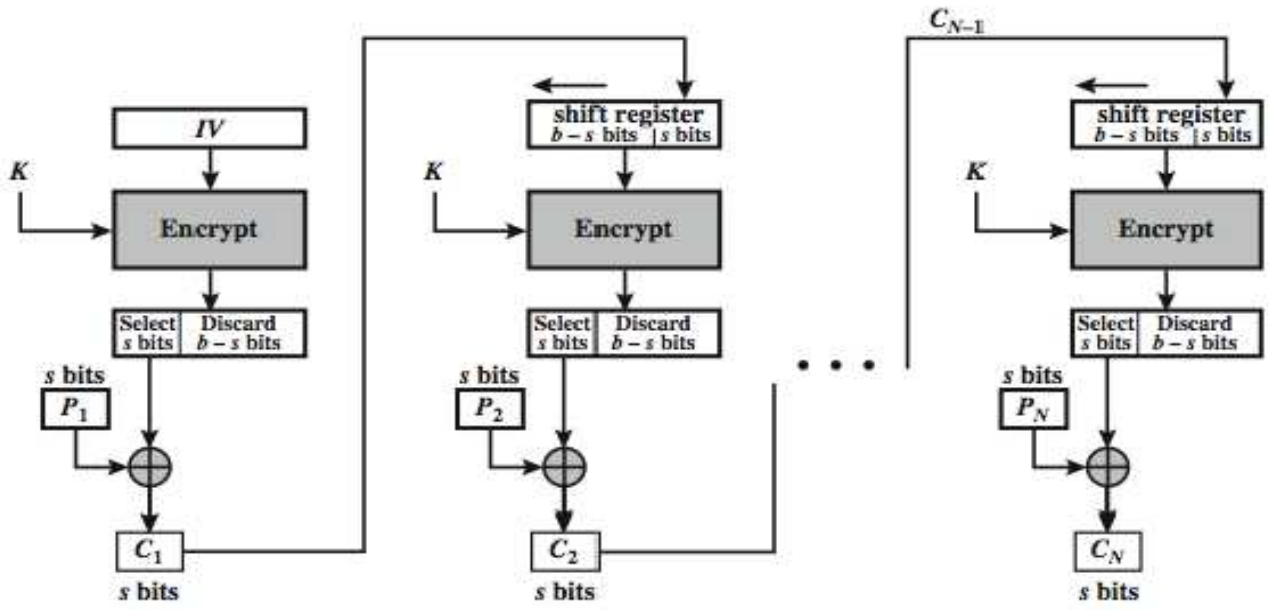
Decryption:

$D(K, C\text{j}) = D(K, E(K, [c\text{j} - 1 \oplus P\text{j}]))$

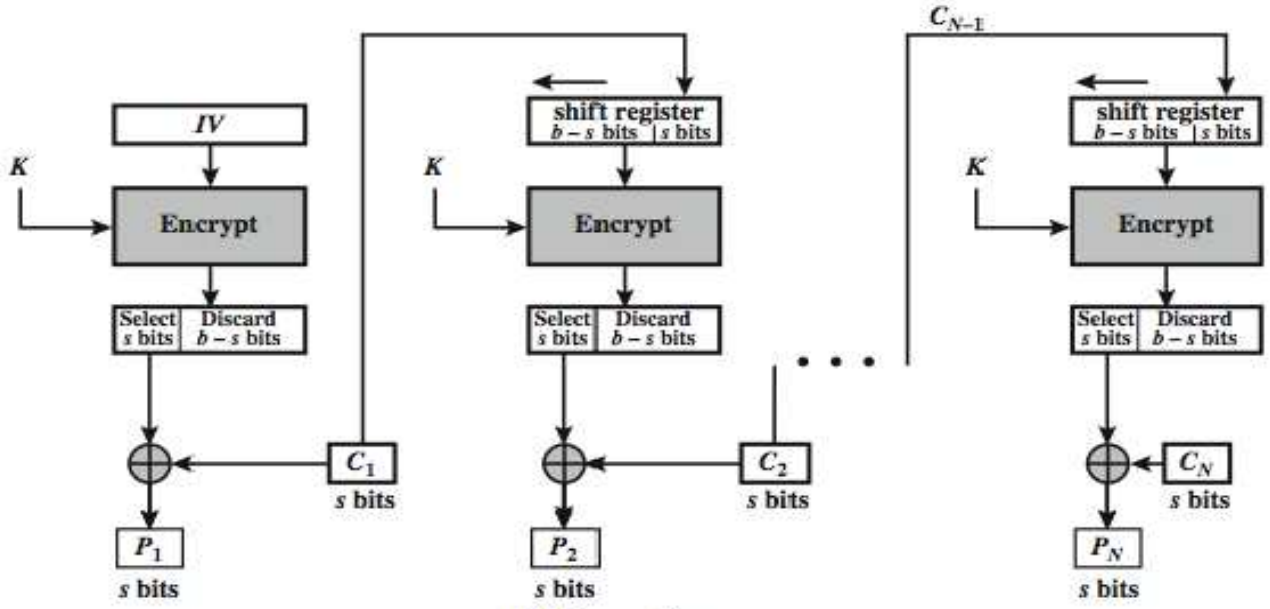
$D(K, C\text{j}) = C\text{j} - 1 \oplus P\text{j}$

$C\text{j} - 1 \oplus D(K, C\text{j}) = C\text{j} - 1 \oplus C\text{j} - 1 \oplus P\text{j} = P\text{j}$

3. Cipher Feedback Mode (CFB)



(a) Encryption

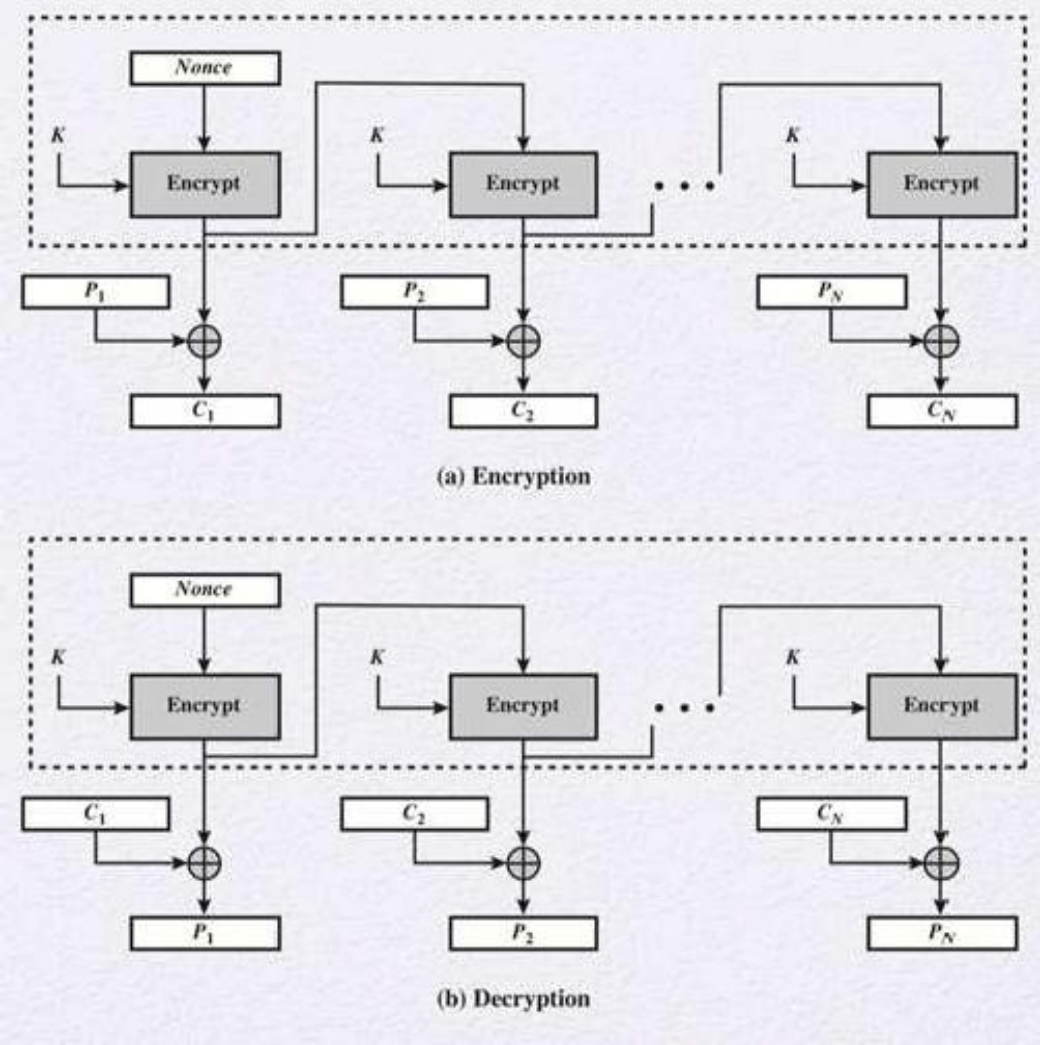


(b) Decryption

- DES is a block cipher, but it may be used as a stream cipher if the Cipher Fee back Mode (CFB) or the Output Feedback Mode (OFB) is used. CFB scheme is depicted below.
- A stream cipher eliminates the need to pad a message to be an integral number of blocks.
- It also can operate in real time.

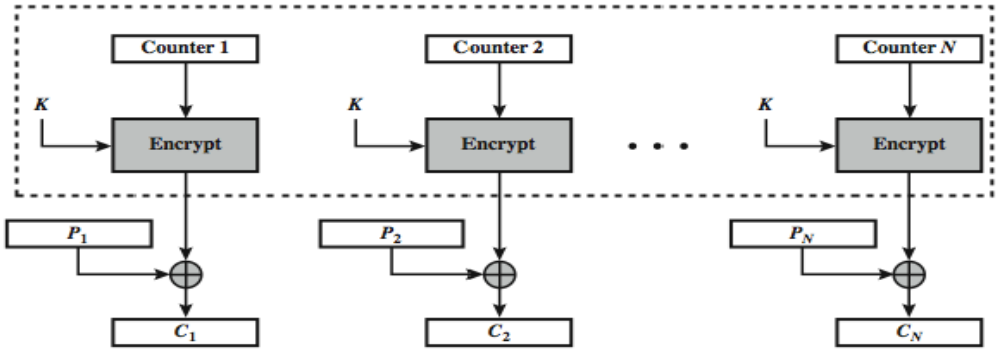
- **'s'** bits is the size usually selected by the user, most of time it is 8 bits.
- In this case, rather than block of 64 bits, the plaintext is divided into segments of s bits.
- **Encryption:** The input to the encryption function is a 64-bit shift register that is initially set to some initialization vector (IV).
- The leftmost (most significant) s bits of the output of the encryption function are X-ORed 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 **s** bits of the shift register.
- This process continues until all plaintext units have been encrypted.
- **Decryption:** The same scheme is used except that the received ciphertext unit is X-ORed with the output of the encryption function to produce the plaintext unit.
- The disadvantage of this scheme is that bit error in one ciphertext propagates to next stage also.

4. Output Feedback Mode (OFB)

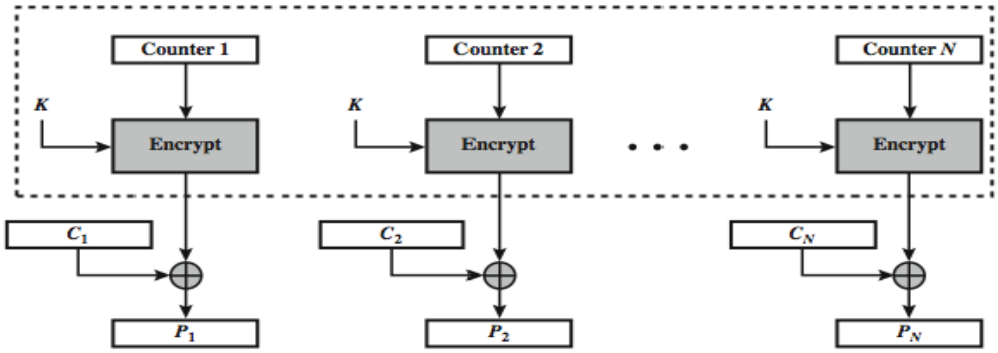


- The Output Feedback Mode (OFB) is similar in structure to that of CFB:
- The difference between CFB and OFB is that in OFB the output of the encryption function is fed back to the shift register in OFB, whereas in CFB the ciphertext is fed to the shift register.
- The other difference is that the OFB mode operates on full blocks of plaintext and ciphertext, not on 's' bit subset.
- One advantage of the OFB method is that bit errors in transmission do not propagate.
- The disadvantage of OFB is that it is more vulnerable to a message stream modification attack than CFB.

5. Counter Mode (CTR)



(a) Encryption



(b) Decryption

- In this mode, a counter equal to the plaintext block size is used.
- The only requirement is 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)
- Counter Mode works as follows:
- **Encryption**: The counter is encrypted and then X-ORed with the plaintext block to produce the cipher text block; there is no chaining.
- **Decryption**: The same sequence of counter values is used. Each encrypted counter is X-ORed with a cipher text block to recover the corresponding plaintext block.
- CTR has following advantages:
  - **Hardware efficiency:** In this mode, encryption (or decryption) can be done in parallel on multiple blocks of plaintext or cipher text. 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, such as aggressive pipelining, multiple instruction dispatch per clock cycle, large number of registers can be effectively utilized.
  - **Preprocessing:** The execution of the encryption algorithm does not depend on input of the plaintext or cipher text. Therefore preprocessing can be used to prepare the output of the encryption boxes which can be fed into the X-OR functions when the plaintext or cipher text input is presented.
  - **Random access:** The **i^{th}** block of plaintext or cipher text can be processed in random-access fashion. With the chaining modes, block cannot be computed until **i– 1** prior block is computed.
  - **Provable security:** It can be shown that CTR is as secure as the other modes.
  - **Simplicity:** CTR mode requires only the implementation of the encryption algorithm and not the decryption algorithm and has a very simple implementation.
- This mode is used in ATM (asynchronous transfer mode) and IPsec (IP security) nowadays.