A modern block cipher is a cipher which encrypts m-bit block of plaintext and decrypts m-bit block of ciphertext. For encryption or decryption, modern block cipher facilitate a K bit key and the decryption algorithm should be inverse of encryption algorithms and for both encryption and decryption similar key is used.

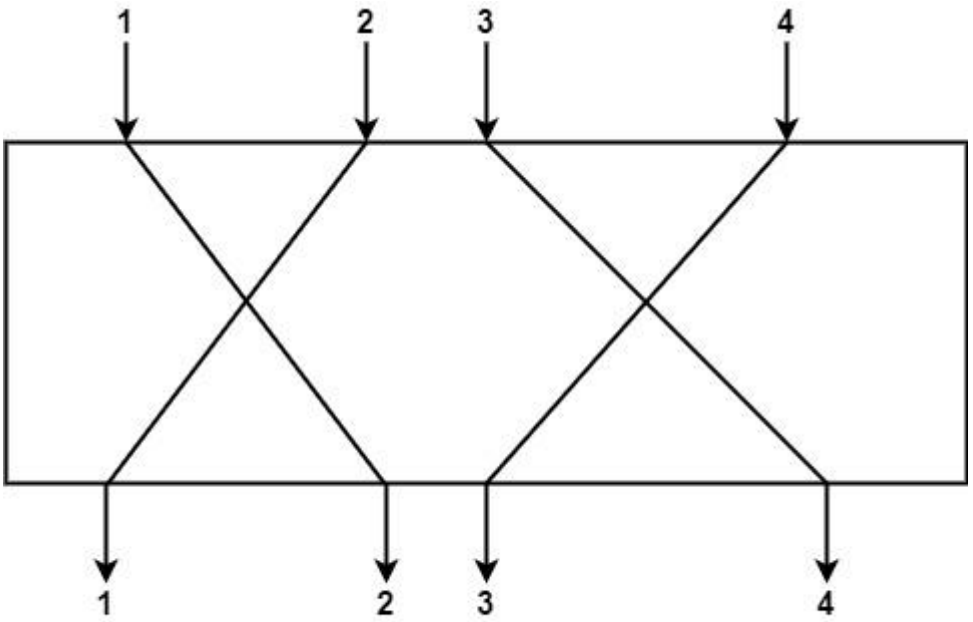
A block cipher works on a plaintext block of n bits to make a cipher text block of n bits. There are possible multiple plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be applicable), each should create a unique cipher text block. Such transformation is known as reversible, or non-singular.

Block cipher modes of operation have been produced to delete the chance of encrypting identical blocks of text the similar method, the ciphertext formed from the previous encrypted block is used to the next block. A block of bits is known as an initialization vector (IV).

There are various components of Modern Block Cipher which are as follows −

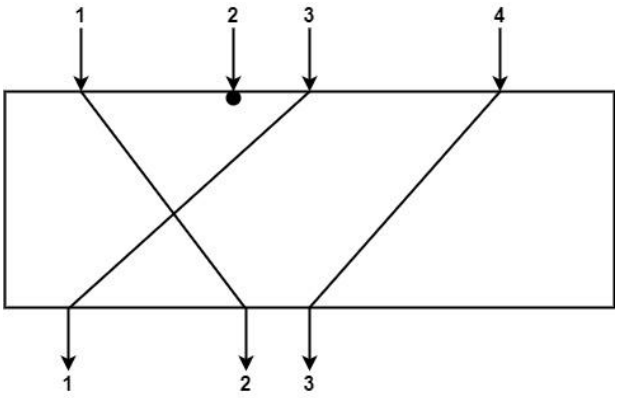
- **D-boxes** − A D-box is a permutation box having similar features as traditional transposition ciphers. D-boxes transpose bits. There are three types of D-boxes which are as follows −

Straight D-box − It creates n inputs, permutes them and supports n outputs. In this, the second input after permutation is the first to be outputted. The first letter in input is permuted to second place, third on fourth place and fourth on third place. There are n! Possible way of mapping in D-box.



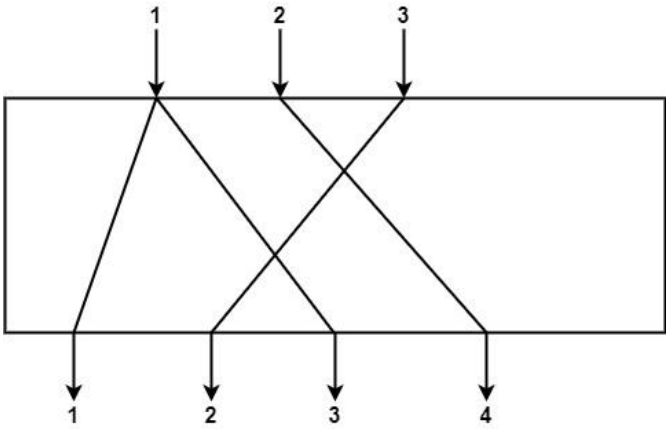
Straight D-box

Compression D-box − This is a D-box with n inputs and m outputs, where m<n. There are various inputs are blocked and do not reach the output. Compression D-boxes are used when it is required to permute bits and at the similar time reduce the number of bits for the next stage.



Compression D-box

Expansion D-box − This is a D-box with n inputs and m outputs, where m >n i.e., there are various inputs are connected to more than one output it is used when it is required to transpose bits and the same increase the multiple bits for the next stage.



Expansion D-box

- **S-boxes** − These are substitution boxes same to the substitution cipher. The input to an S-box can be a n-bit word but the output can be a m-bit word, where m and n are not essentially the same.
- **Circular Shift** − It can also discovered in modern block ciphers, it can be such as leftshift or right-shift. In the circular left shift, shift each bit in n-bit word with m position to the left and the leftmost m-bits are deleted from the left and become the rightmost bits.

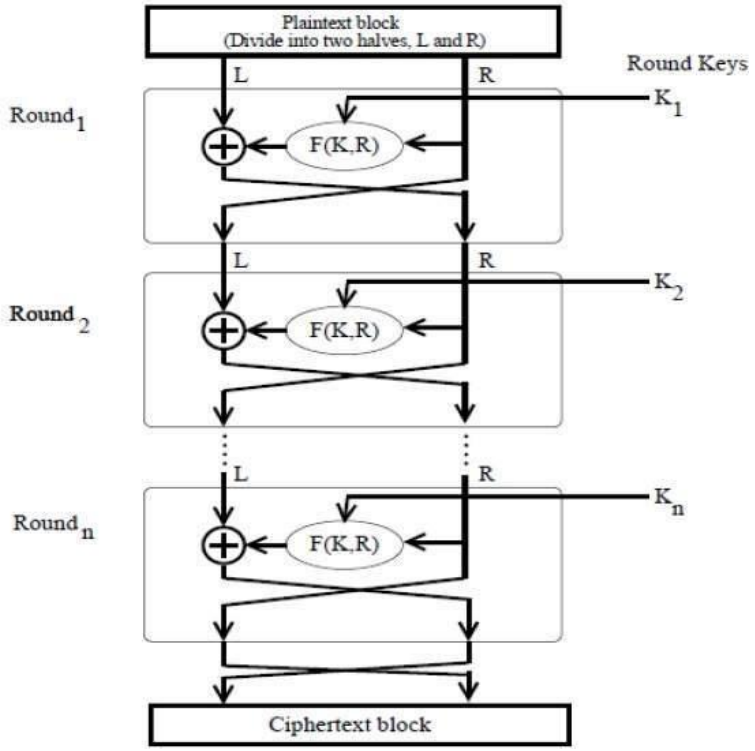
Feistel Block Cipher

Feistel Cipher is not a specific scheme of block cipher. It is a design model from which many different block ciphers are derived. DES is just one example of a Feistel Cipher. A cryptographic system based on Feistel cipher structure uses the same algorithm for both encryption and decryption.

Encryption Process

The encryption process uses the Feistel structure consisting multiple rounds of processing of the plaintext, each round consisting of a “substitution” step followed by a permutation step.

Feistel Structure is shown in the following illustration −



- The input block to each round is divided into two halves that can be denoted as L and R for the left half and the right half.
- In each round, the right half of the block, R, goes through unchanged. But the left half, L, goes through an operation that depends on R and the encryption key. First, we apply an encrypting function 'f' that takes two input − the key K and R. The function produces the output f(R,K). Then, we XOR the output of the mathematical function with L.
- In real implementation of the Feistel Cipher, such as DES, instead of using the whole encryption key during each round, a round-dependent key (a subkey) is derived from the encryption key. This means that each round uses a different key, although all these subkeys are related to the original key.
- The permutation step at the end of each round swaps the modified L and unmodified R. Therefore, the L for the next round would be R of the current round. And R for the next round be the output L of the current round.
- Above substitution and permutation steps form a 'round'. The number of rounds are specified by the algorithm design.
- Once the last round is completed then the two sub blocks, 'R' and 'L' are concatenated in this order to form the ciphertext block.

The difficult part of designing a Feistel Cipher is selection of round function 'f'. In order to be unbreakable scheme, this function needs to have several important properties that are beyond the scope of our discussion.

Block Cipher Design Principles

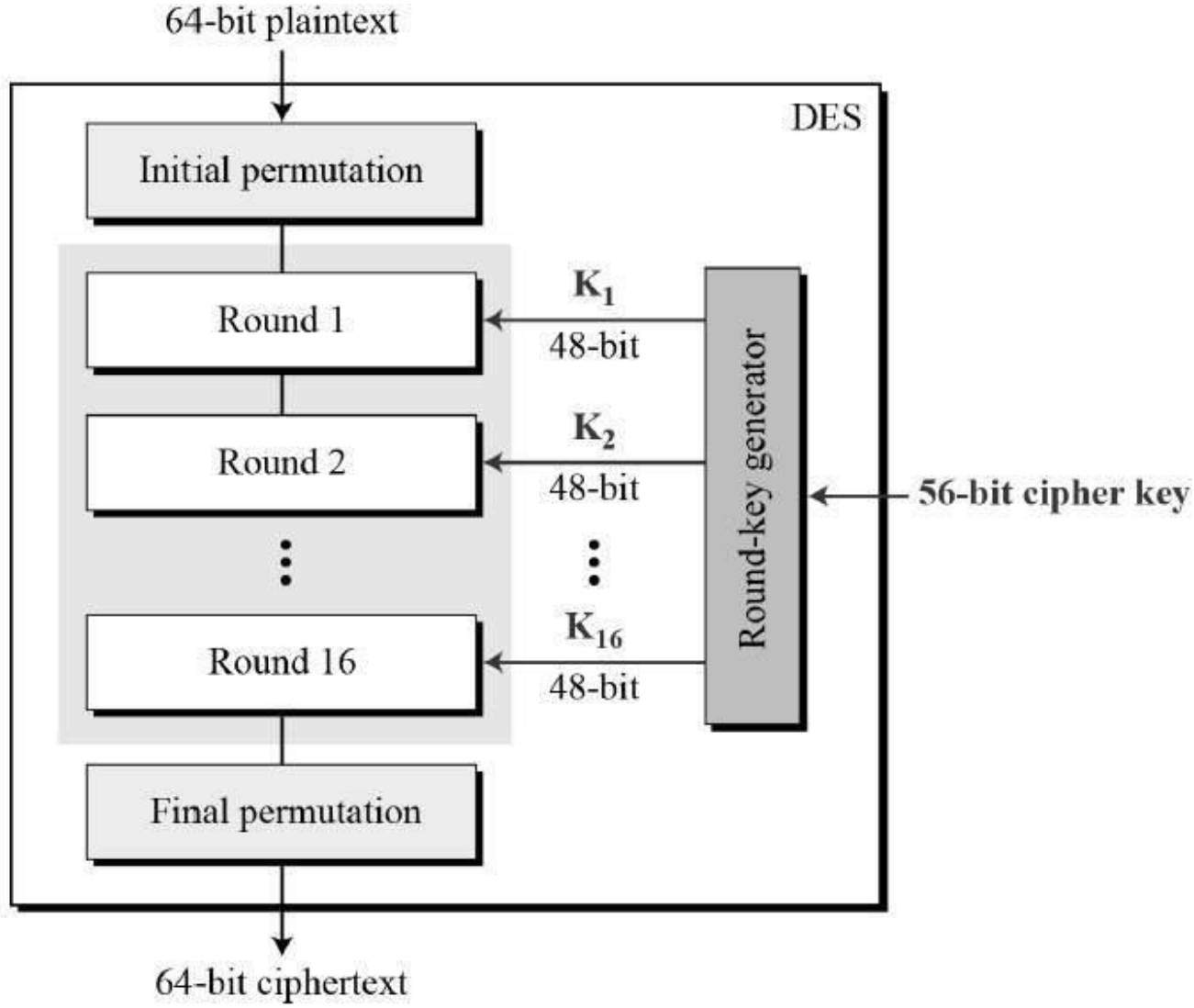
Block ciphers are built in the Feistel cipher structure. Block cipher has a specific number of rounds and keys for generating ciphertext. For defining the complexity level of an algorithm few design principles are to be considered.
These are explained as following below :

1. **Number of Rounds –**
   The number of Rounds is regularly considered in design criteria, it just reflects the number of rounds to be suitable for an algorithm to make it more complex, in DES we have 16 rounds ensuring it to be more secure while in AES we have 10 rounds which makes it more secure.
2. **Design of function F –**
   The core part of the Feistel Block cipher structure is the Round Function. The complexity of cryptanalysis can be derived from the Round function i.e. the increasing level of complexity for the round function would be greatly contributing to an increase in complexity.
   To increase the complexity of the round function, the avalanche effect is also included in the round function, as the change of a single bit in plain text would produce a mischievous output due to the presence of avalanche effect.
3. **Key schedule algorithm –**
   In Feistel Block cipher structure, each round would generate a sub-key for increasing the complexity of cryptanalysis. The Avalanche effect makes it more complex in deriving sub-key. Decryption must be done very carefully to get the actual output as the avalanche effect is present in it.

DES

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration −

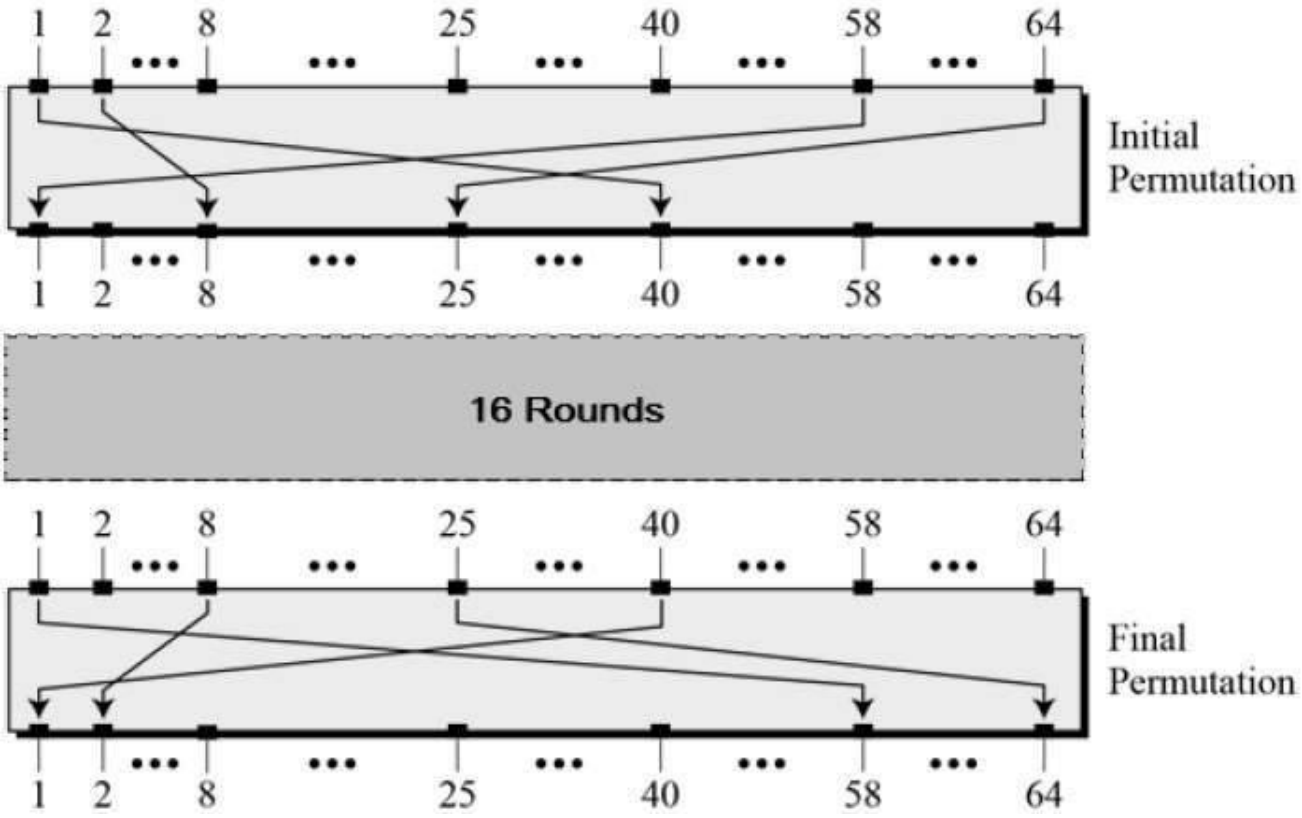


Since DES is based on the Feistel Cipher, all that is required to specify DES is −

- Round function
- Key schedule
- Any additional processing − Initial and final permutation

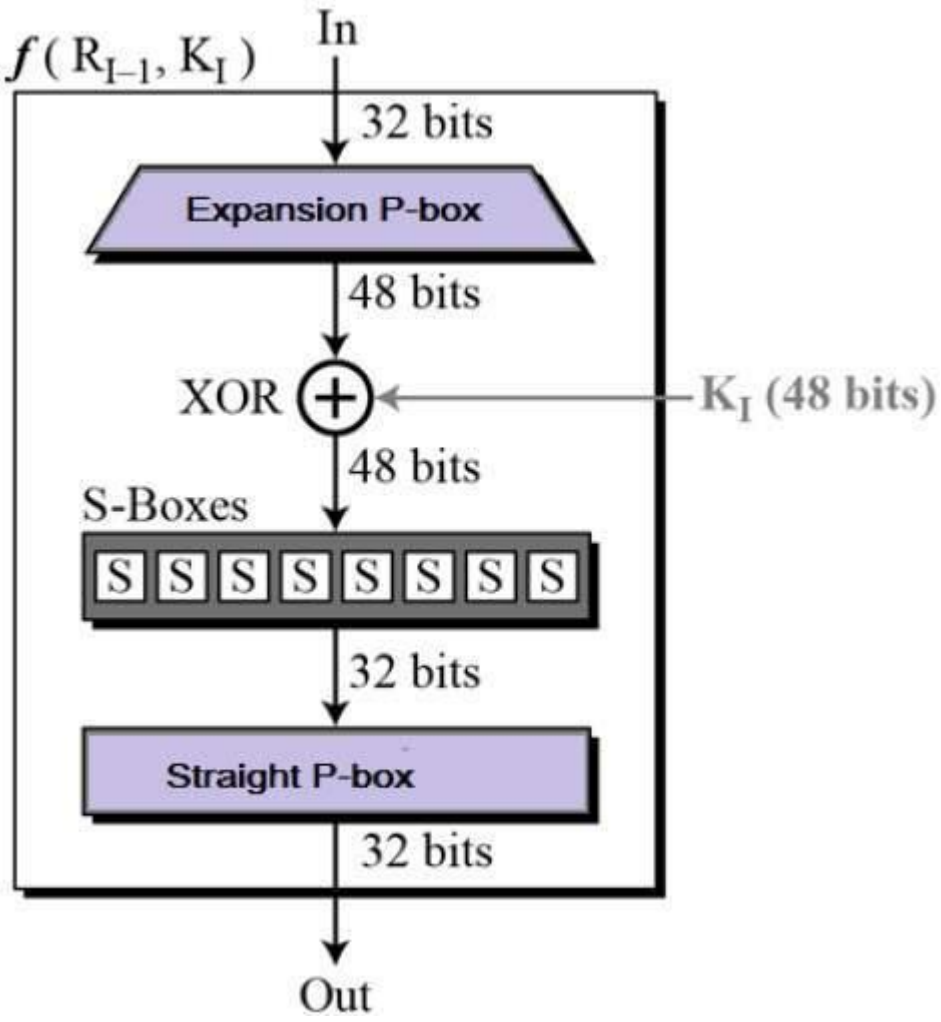
Initial and Final Permutation

The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES. The initial and final permutations are shown as follows −

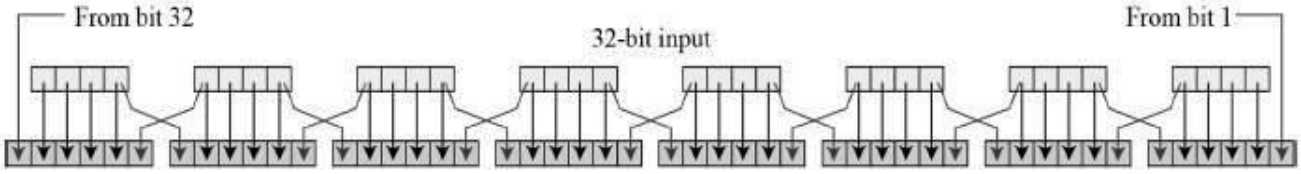


Round Function

The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



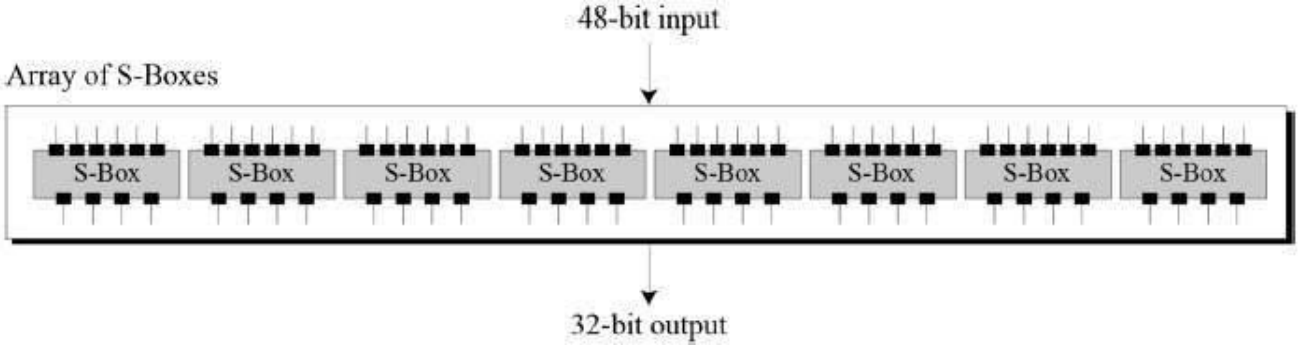
- **Expansion Permutation Box** − Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits. Permutation logic is graphically depicted in the following illustration −



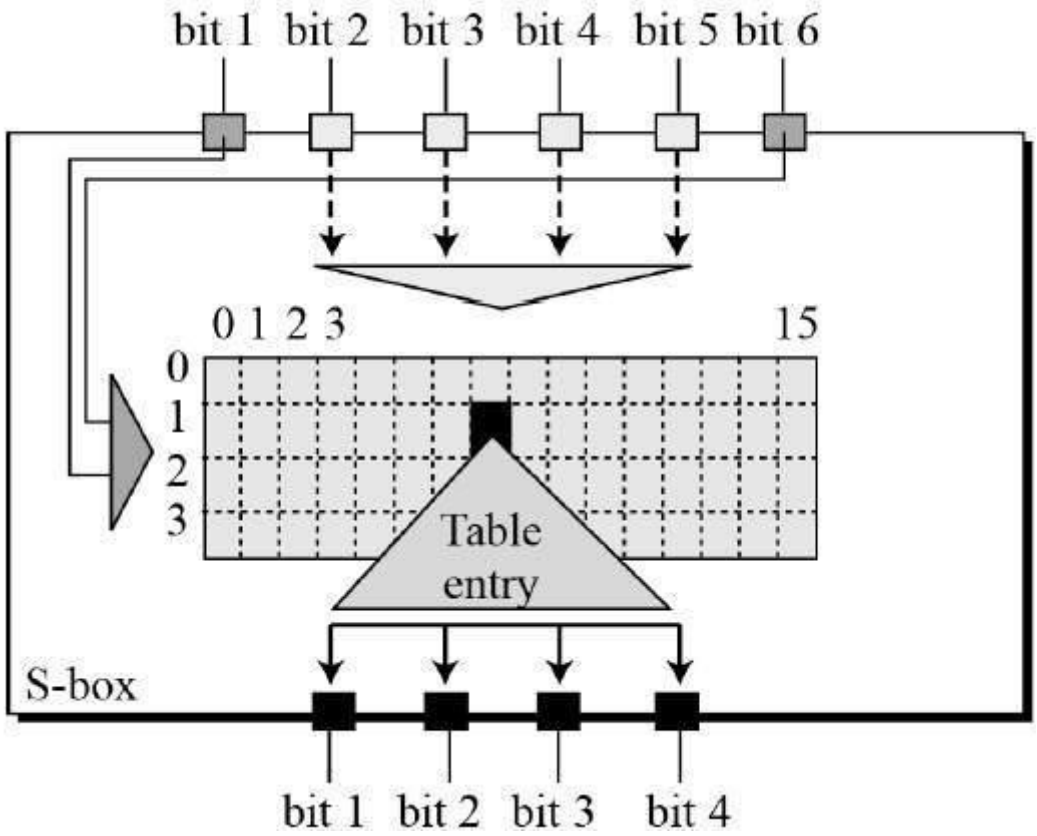
- The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown −

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

- **XOR (Whitener).** − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.
- **Substitution Boxes.** − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration −



- The S-box rule is illustrated below −

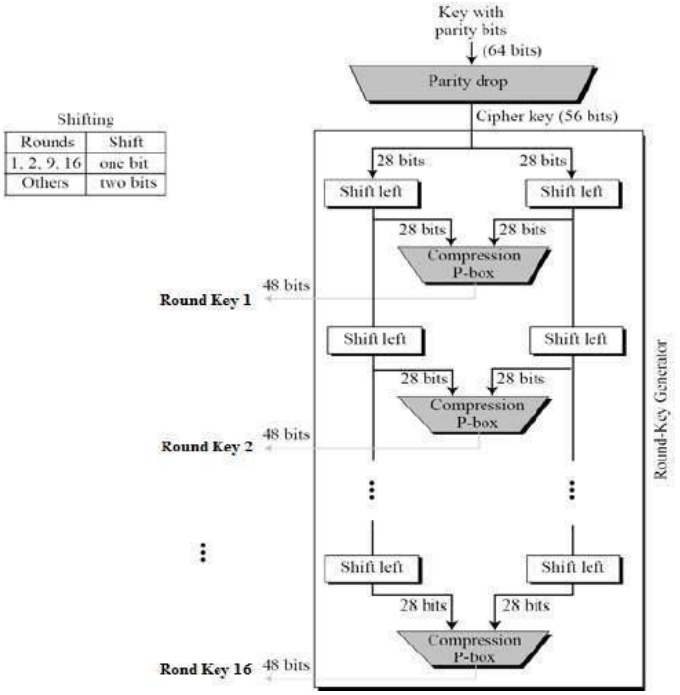


- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.
- **Straight Permutation** − The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

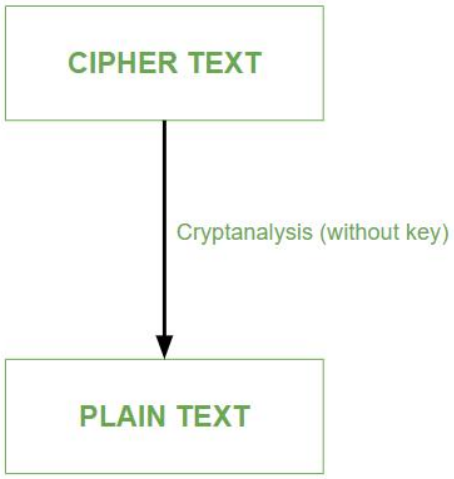
Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration −



The logic for Parity drop, shifting, and Compression P-box is given in the DES description.

Cryptanalysis is the process of transforming or decoding communications from non-readable to readable format without having access to the real key. OR we may say it is the technique of retrieving the plain text of the communication without having access to the key. *Cryptoanalysis is the art, science, or practice of decrypting encrypted messages.* The secret key used for encryption and decoding is considered to be unknown to the cryptologists, mathematicians, and other scientists participating in the process. In contrast to a brute force attack, this form of analysis seeks vulnerabilities in a cryptosystem.
Cryptanalysis frequently comprises a direct evaluation of the cryptosystem in use, which is essentially an advanced concentrated mathematical attempt at decryption utilizing knowledge about the encryption scheme that is already available. They can employ intercepted encrypted messages (ciphertext), intercepted complete, partial, likely, or similar original messages (plaintext), or information (encrypted or original) that is known to be used adaptively in subsequent trials.



Process of cryptanalysis

Cryptanalysis is used to break cryptographic security systems and gain access to the contents of the encrypted messages, even if the cryptographic key is unknown.

Types of Cryptanalytic Attacks:

1. **Ciphertext only attack:**
    1. In this type of cryptanalytic attack, the attacker has the knowledge of only the ciphertext.
    2. The attacker has to detect the plain text using the ciphertext only.
    3. This type of attack is not very easy to be implemented.
2. **Known plain text only attack:**
    1. In this type of cryptanalytic attack, the attacker has the knowledge of some plain text as well as ciphertext.
    2. The attacker tries to decrypt the messages using these two.
    3. This type of attack is somewhat easy to implement.

Different Forms of Cryptanalysis:

Cryptanalysis basically has two forms:

1. Linear Cryptanalysis:

Linear cryptanalysis is a general type of cryptanalysis based on discovering affine approximations to a cipher's action in cryptography. Block and stream ciphers have both

been subjected to attacks. Linear cryptanalysis is one of the two most common attacks against block ciphers, with differential cryptanalysis being the other.

2. **Differential Cryptanalysis:**

Differential cryptanalysis is a sort of cryptanalysis that may be used to decrypt both block and stream ciphers, as well as cryptographic hash functions. In the widest sense, it is the study of how alterations in information intake might impact the following difference at the output. In the context of a block cipher, it refers to a collection of strategies for tracking differences across a network of transformations, finding where the cipher displays non-random behavior, and using such attributes to recover the secret key (cryptography key).

Introduction to Block Cipher modes

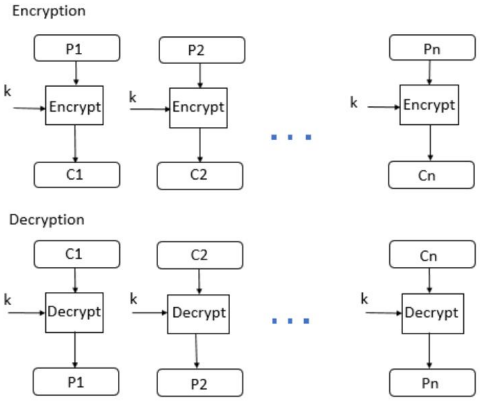
There are five types of operations in block cipher modes, ECB (Electronic Code Block) mode, CBC (Cipher Block Chaining) mode, CFB (Cipher Feedback) mode, OFB (Output Feedback) mode and CTR (Counter) mode. Where ECB and CBC mode works on block ciphers, and CFB and OFB mode works on block ciphers acting as stream ciphers. ECB is used for transmitting a single value insecure manner, CBC is used for encrypting blocks of text authentication, CFB is used for transmitting an encrypted stream of data authentication, OFB is used for transmitting an encrypted stream of data, CTR is used for transmitting block-oriented applications.

Block cipher modes of operation

There are 5 modes of operation in the block cipher.

1. ECB mode

- ECB mode stands for Electronic Code Block Mode. It is one of the simplest modes of operation. In this mode, the plain text is divided into a block where each block is 64 bits. Then each block is encrypted separately. The same key is used for the encryption of all blocks. Each block is encrypted using the key and makes the block of ciphertext.
- At the receiver side, the data is divided into a block, each of 64 bits. The same key which is used for encryption is used for decryption. It takes the 64-bit ciphertext and, by using the key convert the ciphertext into plain text.
- As the same key is used for all blocks' encryption, if the block of plain text is repeated in the original message, then the ciphertext's corresponding block will also repeat. As the same key used for tor all block, to avoid the repetition of block ECB mode



is used for an only small message where the repetition of the plain text block is less.

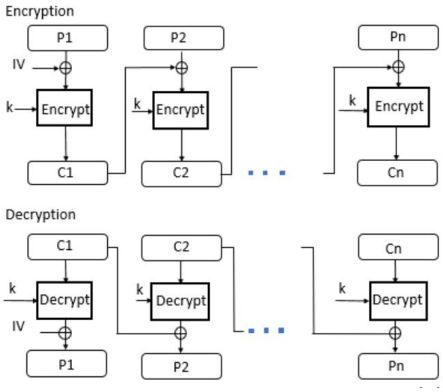
2. CBC Mode

- CBC Mode stands for Cipher block Mode at the sender side; the plain text is divided into blocks. In this mode, IV(Initialization Vector) is used, which can be a random block of text. IV is used to make the ciphertext of each block unique.

- The first block of plain text and IV is combined using the XOR operation and then encrypted the resultant message using the key and form the first block of ciphertext. The first block of ciphertext is used as IV for the second block of plain text. The same procedure will be followed for all blocks of plain text.
- At the receiver side, the ciphertext is divided into blocks. The first block ciphertext is decrypted using the same key, which is used for encryption. The decrypted result will be XOR with the IV and form the first block of plain text. The second block of ciphertext is also decrypted using the same key, and the result of the decryption will be XOR with the first block of ciphertext and form the second block of plain text. The same procedure is used for all the blocks.

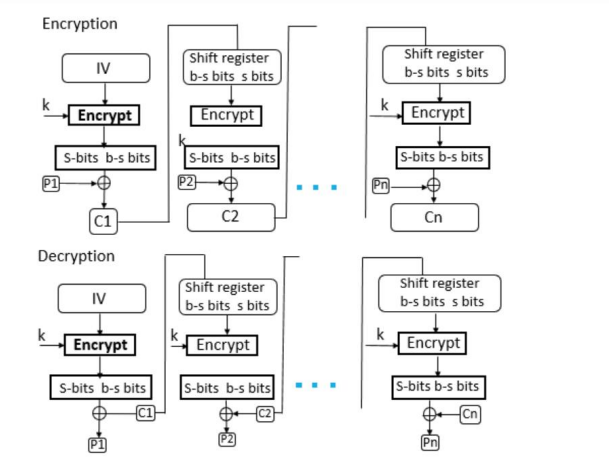
- CBC Mode ensures that if the block of plain text is repeated in the original message, it will produce a different ciphertext for corresponding blocks.

  Note that the key which is used in CBC mode is the same; only the IV is different, which is initialized at a starting point.



3. CFB Mode

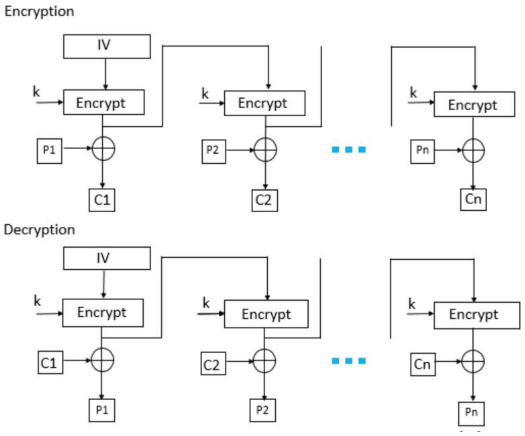
- CFB mode stands for Cipher Feedback Mode. In this mode, the data is encrypted in the form of units where each unit is of 8 bits.
- Like cipher block chaining mode, IV is initialized. The IV is kept in the shift register. It is encrypted using the key and form the ciphertext.
- Now the leftmost j bits of the encrypted IV is XOR with the plain text's first j bits. This process will form the first part of the ciphertext, and this ciphertext will be transmitted to the receiver.
- Now the bits of IV is shifted left by j bit. Therefore the rightmost j position of the shift register now has unpredictable data. These rightmost j positions are now filed with the ciphertext. The process will be repeated for all plain text units.

4. 

OFB mode

- OFB Mode stands for output feedback Mode. OFB mode is similar to CDB mode; the only difference is in CFB, the ciphertext is used for the next stage of the encryption process, whereas in OFB, the output of the IV encryption is used for the next stage of the encryption process.
- The IV is encrypted using the key and form encrypted IV. Plain text and leftmost 8 bits of encrypted IV are combined using XOR and produce the ciphertext.

- For the next stage, the ciphertext, which is the form in the previous stage, is used as an IV for the next iteration. The same procedure is followed for all blocks.

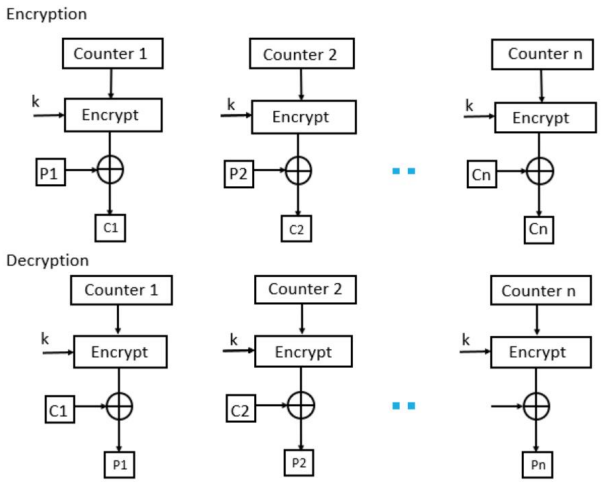


5. CTR Mode

- CTR Mode stands for counter mode. As the name is counter, it uses the sequence of numbers as an input for the algorithm. When the block is encrypted, to fill the next register next counter

value is used.

Note: the counter value will be incremented by 1.



- For encryption, the first counter is encrypted using a key, and then the plain text is XOR with the encrypted result to form the ciphertext.
- The counter will be incremented by 1 for the next stage, and the same procedure will be followed for all blocks. For decryption, the same sequence will be used. Here to convert ciphertext into plain text, each ciphertext is XOR with the encrypted counter. For the next stage, the counter will be incremented by the same will be repeated for all Ciphertext blocks.

AES

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

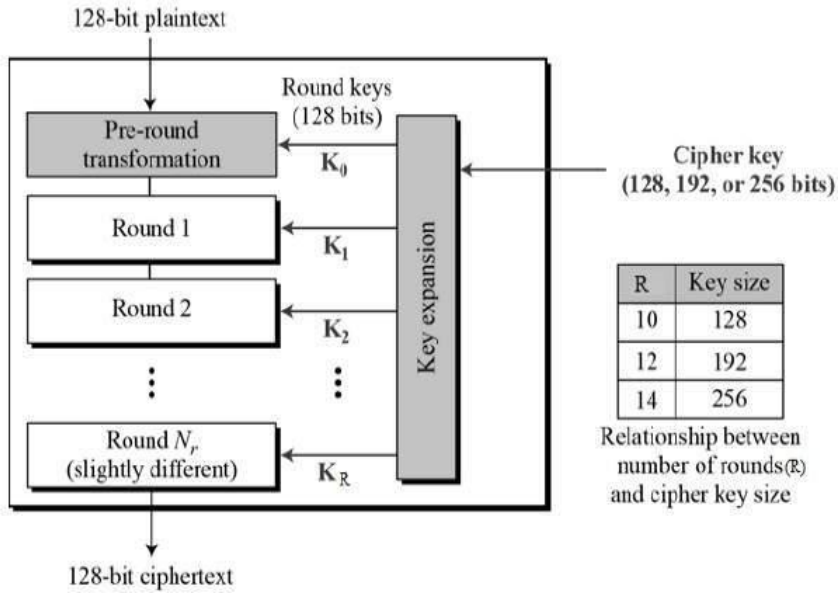
Operation of AES

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration −

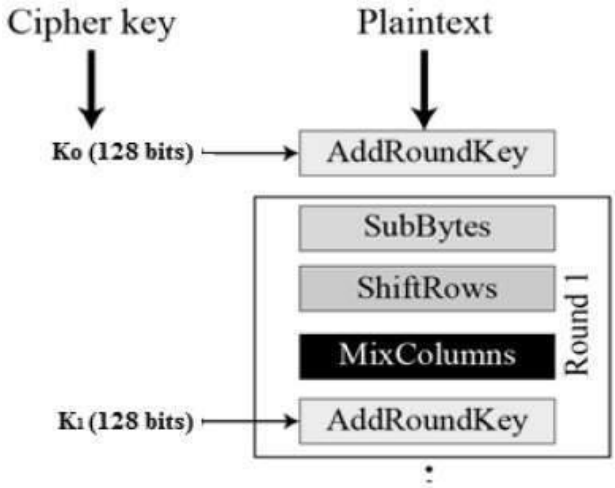


R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below –



Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows −

- First row is not shifted.

- Second row is shifted one (byte) position to the left.
- Third row is shifted two positions to the left.
- Fourth row is shifted three positions to the left.
- The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.