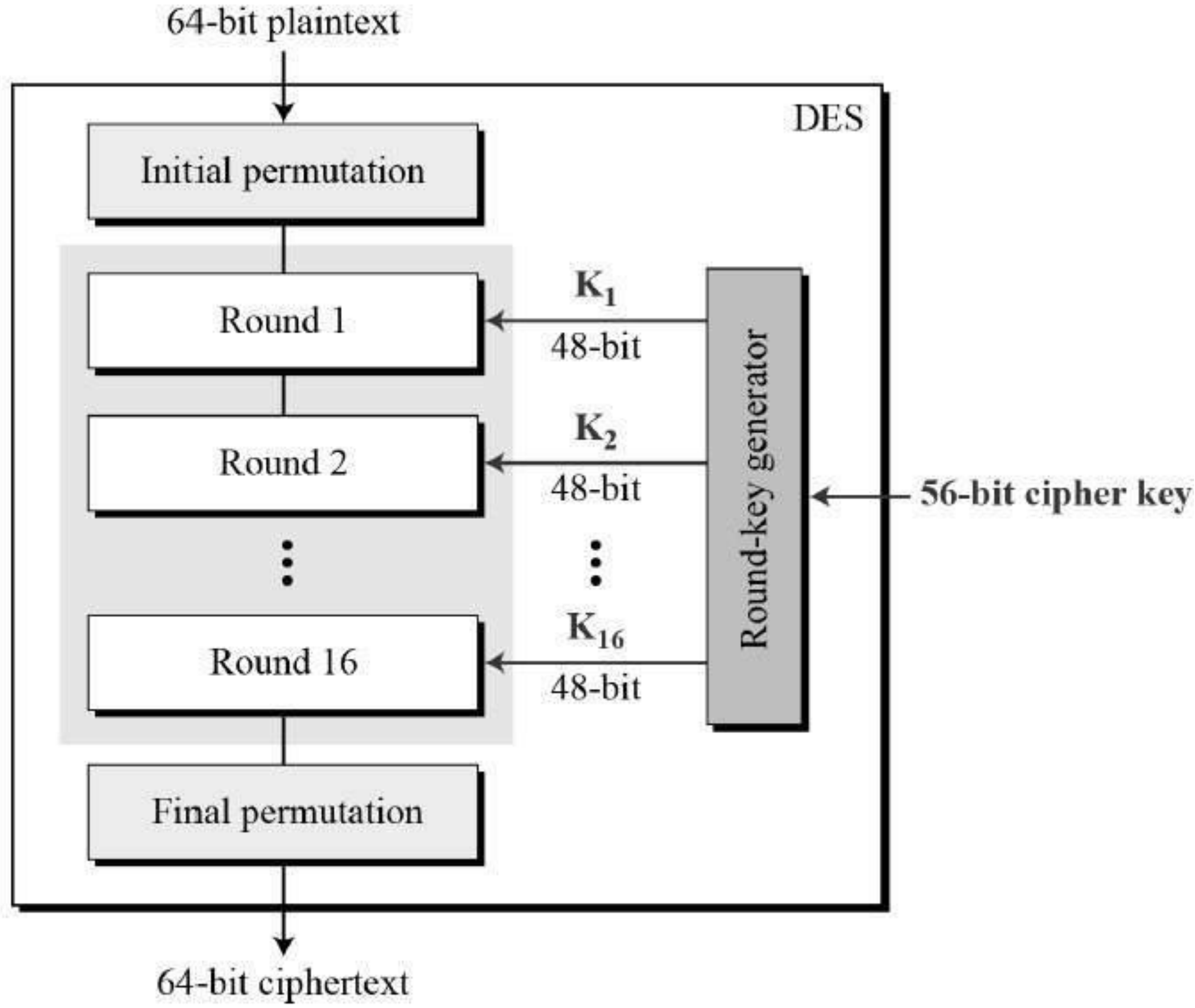
The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

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 −

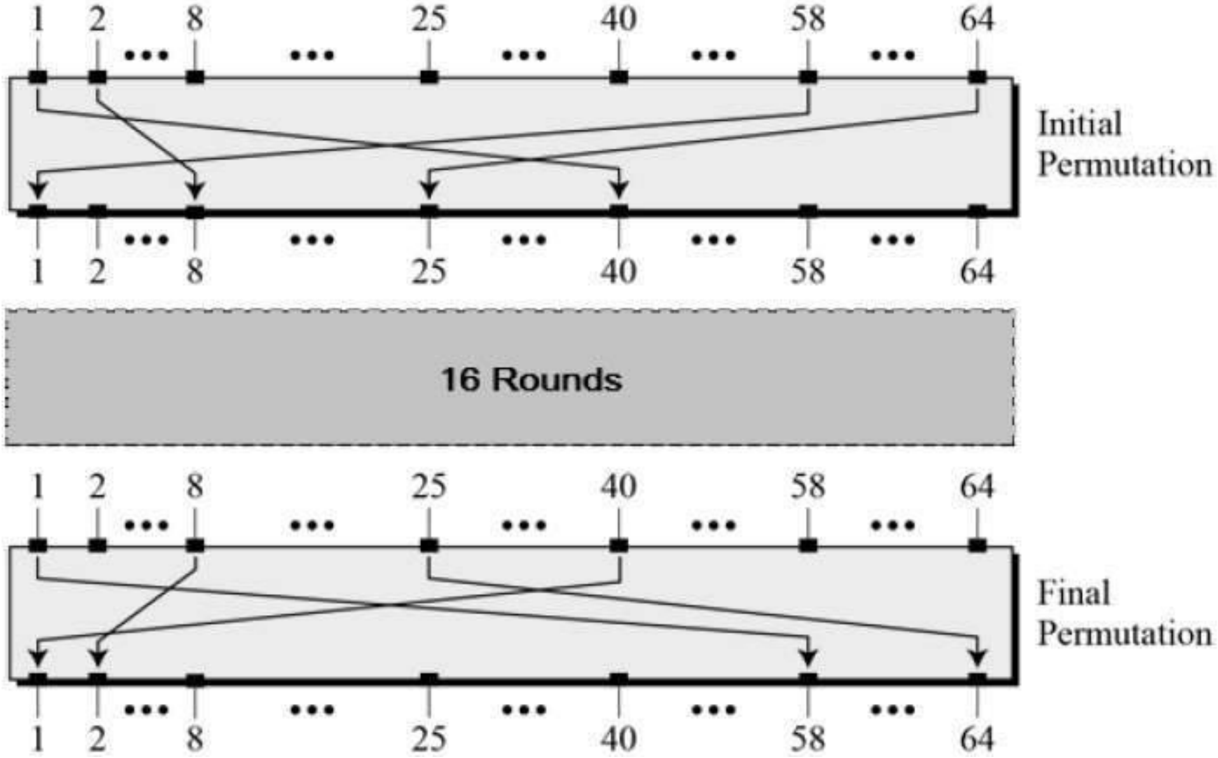


Since DES is based on the Feistel Cipher, all that is required to specify DES is −

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

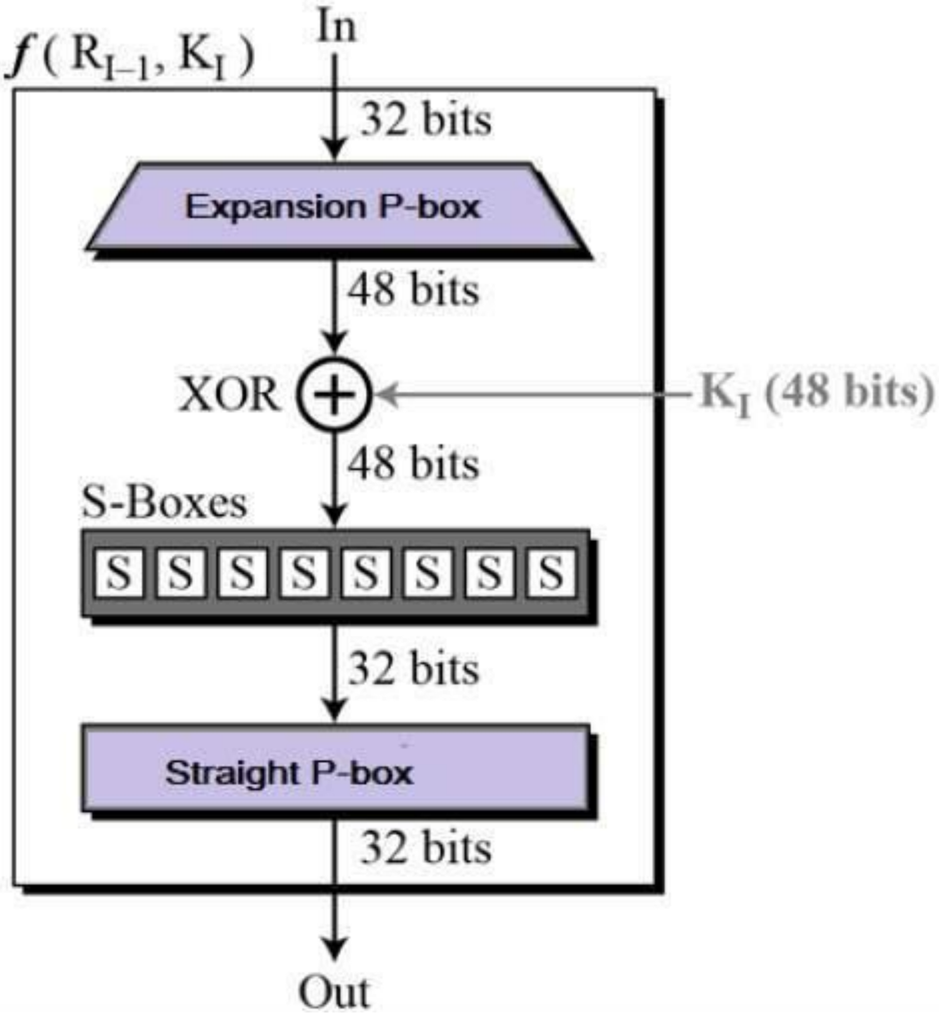
Initial and Final Permutation

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 –

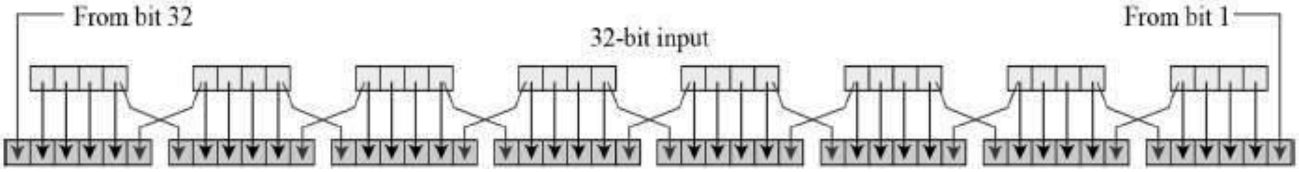


Round Function

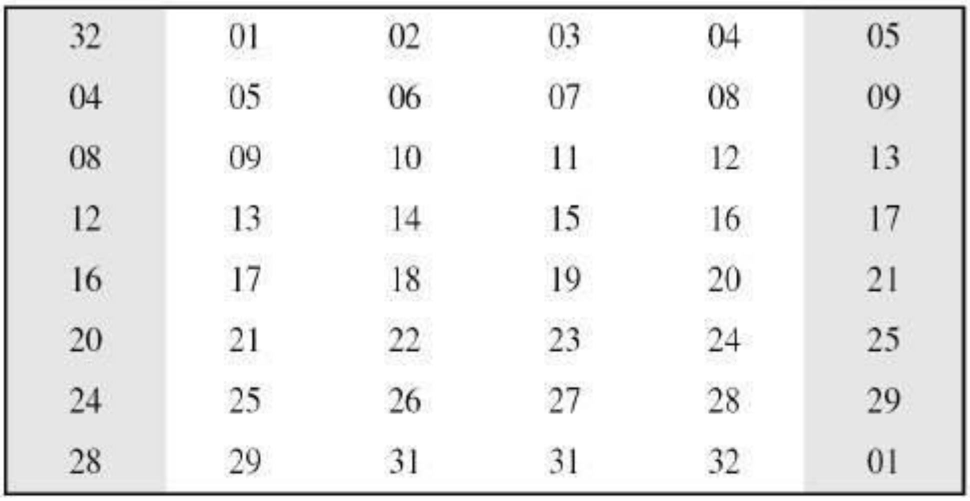
The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



- **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 –

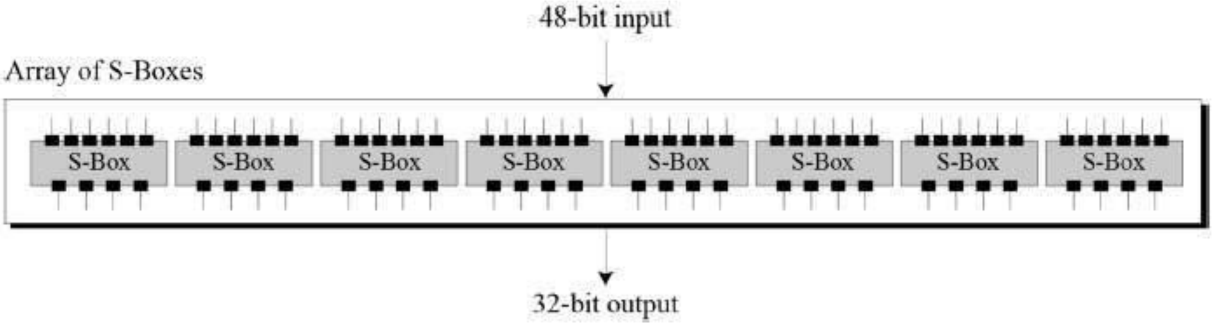


- The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown −

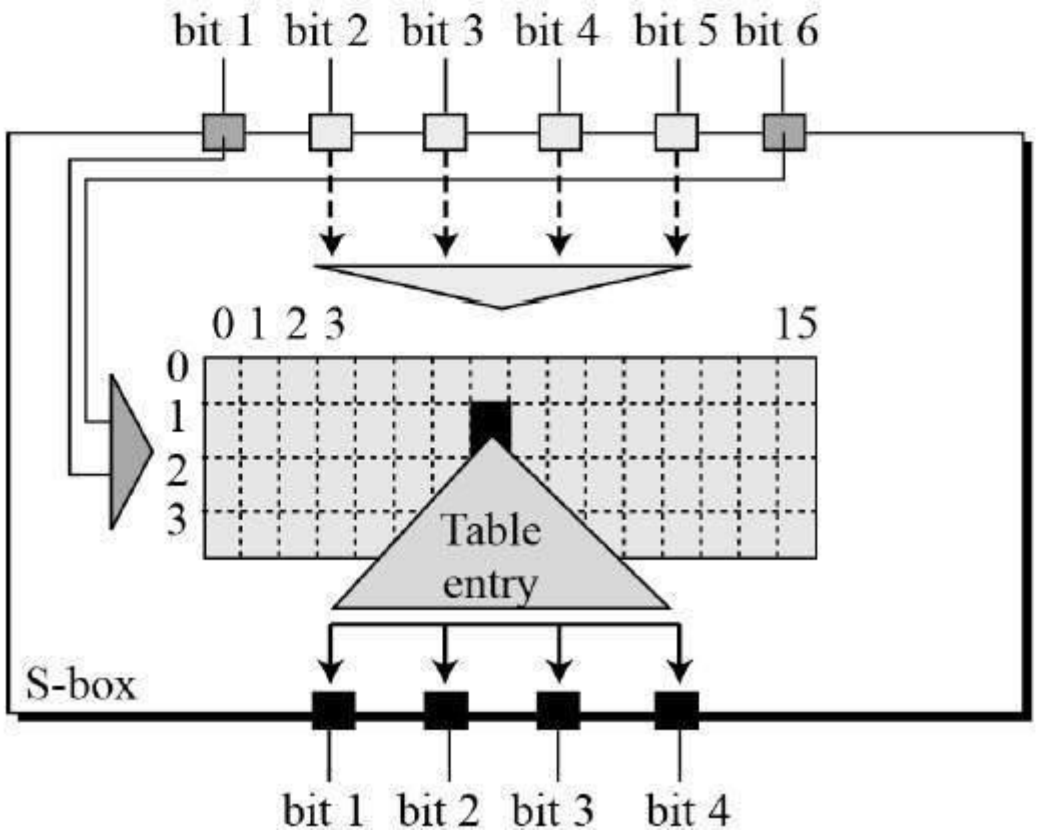


32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

- **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 −

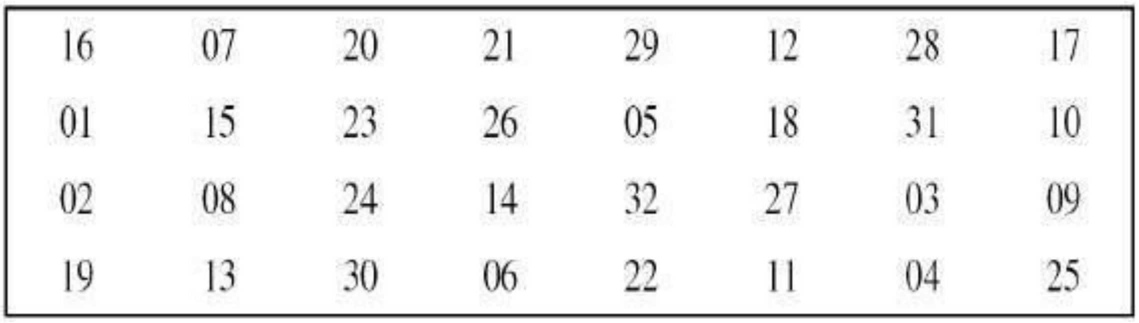


The S-box rule is illustrated below –



- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.

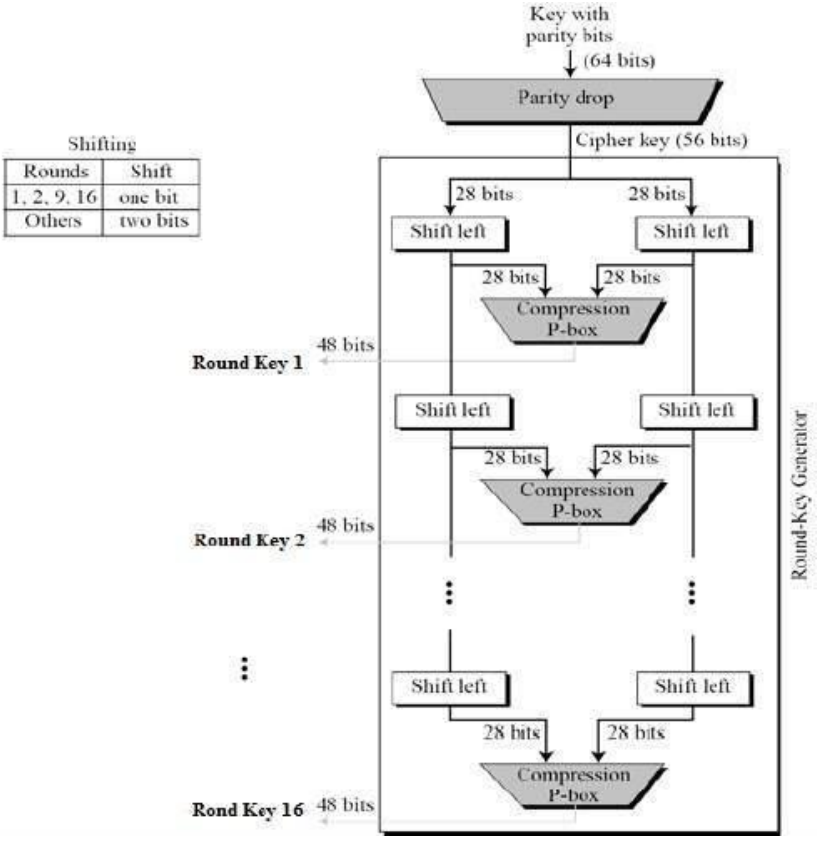
- **Straight Permutation** − The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:



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 −



Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

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

DES Analysis

The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.

- **Avalanche effect −** A small change in plaintext results in the very great change in the ciphertext.
- **Completeness −** Each bit of ciphertext depends on many bits of plaintext.

During the last few years, cryptanalysis have found some weaknesses in DES when key selected are weak keys. These keys shall be avoided.

DES has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on DES other than exhaustive key search.

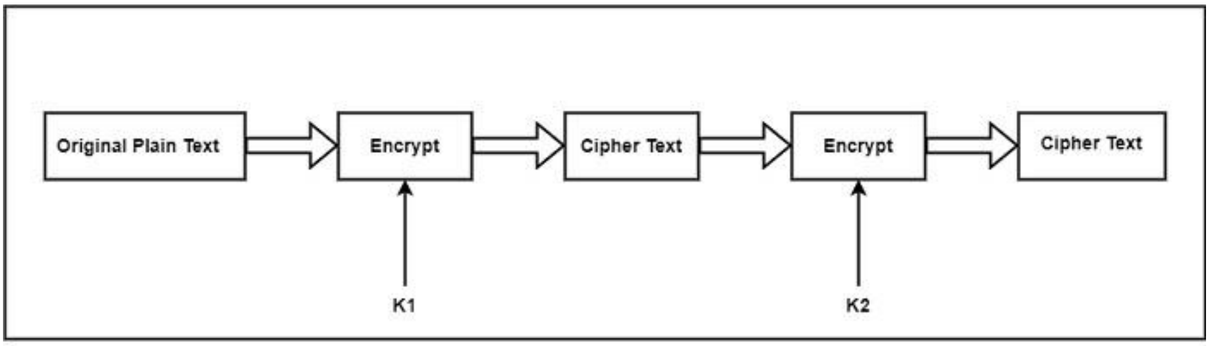
DOUBLE DES

The Data Encryption Standard (DES) is a symmetric key block cipher which creates 64-bit plaintext and 56-bit key as an input and makes 64-bit cipher text as output. The DES function is create up of P and S-boxes. P-boxes transpose bits and S-boxes substitute bits to make a cipher.

DES is a Feistel Block Cipher implementation, called a LUCIFER. It need a Feistel structure with 16 rounds, where a different key can be used for each round. The major reasons to understand DES (Data Encryption Standard) is that it forms the foundation for encryption algorithms. This creates it easy for one to learn the implementation or operating of currently used encryption algorithms or methods, which are much quicker than the DES algorithm.

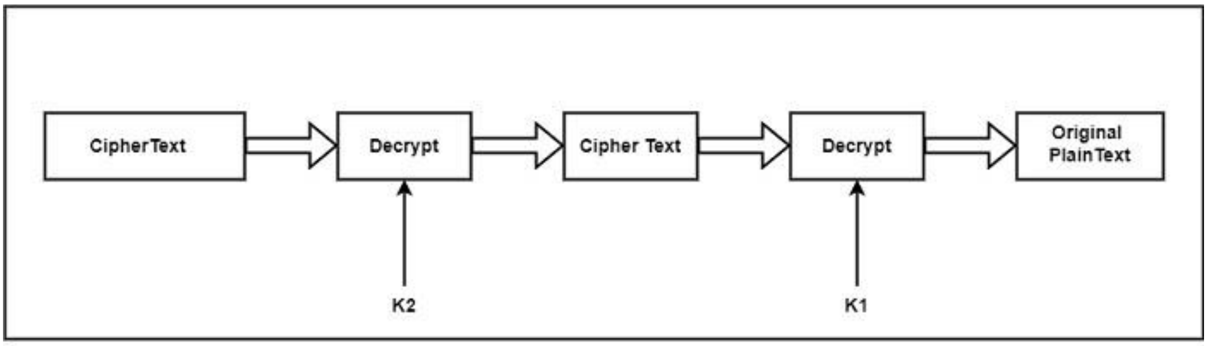
Double DES is an encryption approach which uses two example of DES on same plain text. In both examples it provides different keys to encode the plain text. Double DES is easily to learn.

Double DES uses two keys, such as k1and k2. It can implement DES on the original plain text using k1 to get the encrypted text. It can implement DES on the encrypted text, but this time with the different key k2. The final output is the encryption of encrypted text as shown in the figure.



Double DES Encryption

The double encrypted cipher-text block is first decrypted using the key K2 to make the singly encrypted cipher text. This ciphertext block is then decrypted using the key K1 to acquire the original plaintext block.



Double DES Decryption

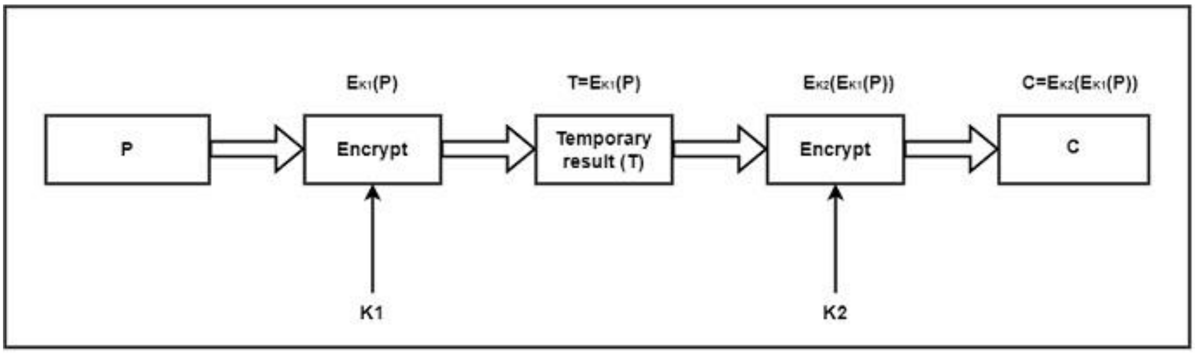
If it can use a key of only 1 bit, there are two possible keys including 0 and 1. If it can use a 2 bit key, there are four possible key values such as (00, 01, 10 and 11).

In general, if it can use an n-bit key, the cryptanalyst has to implement 2^n operations to try out all the possible keys. If it can use two different keys, each including n bits, the cryptanalyst would require 2^{2n} attempt to crack the key.

Double DES needed a key search of (2^{2*56}), i. e. , 2^{112}keys. It introduce the terms of the meet-in-the-middle attack. This attack contains encryption from one end, decryption from the other and connecting the results in the middle.

Consider that the cryptanalyst understand two basic pieces of information including P (a plain-text block) and C (the corresponding final cipher-text block) for a message. The numerical expression of Double DES as shown in the figure.

The result of the first encryption is known as T and is indicated as T = E_{k1}(P) [i.e., encrypt the block P with key K1]. After this encrypted block is encrypted with another key K2, it indicate the result as C = $E_{K2}(E_{K1}(P))$ [i.e., encrypt the already encrypted block T, with a different key K2, and call the final ciphertext as C].



Mathematical Expression of Double DES

TRIPLE DES

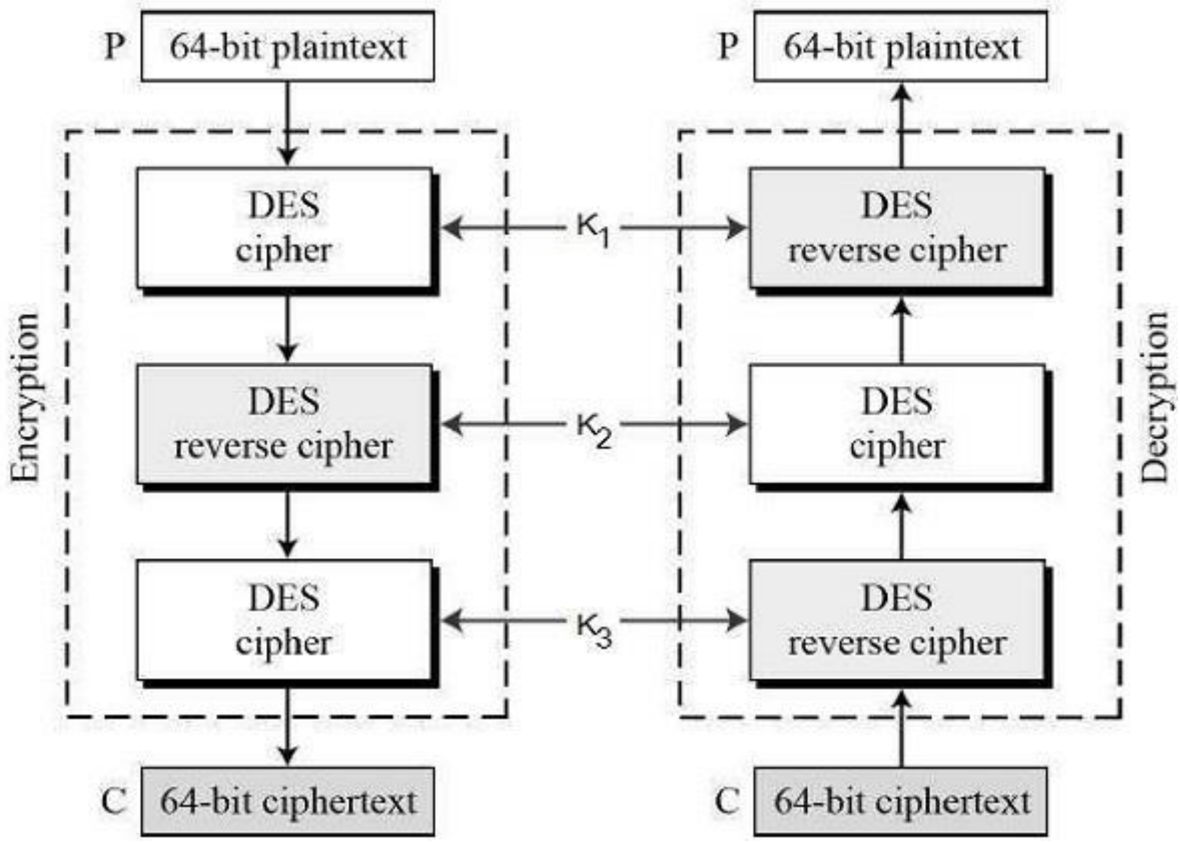
The speed of exhaustive key searches against DES after 1990 began to cause discomfort amongst users of DES. However, users did not want to replace DES as it takes an enormous amount of time and money to change encryption algorithms that are widely adopted and embedded in large security architectures.

The pragmatic approach was not to abandon the DES completely, but to change the manner in which DES is used. This led to the modified schemes of Triple DES (sometimes known as 3DES).

Incidentally, there are two variants of Triple DES known as 3-key Triple DES (3TDES) and 2-key Triple DES (2TDES).

3-KEY Triple DES

Before using 3TDES, user first generate and distribute a 3TDES key K, which consists of three different DES keys K_1, K_2 and K_3. This means that the actual 3TDES key has length 3×56 = 168 bits. The encryption scheme is illustrated as follows −



The encryption-decryption process is as follows −

- Encrypt the plaintext blocks using single DES with key K_1.
- Now decrypt the output of step 1 using single DES with key K_2.
- Finally, encrypt the output of step 2 using single DES with key K_3.
- The output of step 3 is the ciphertext.

- Decryption of a ciphertext is a reverse process. User first decrypt using $K_{3,}$ then encrypt with $K_{2,}$ and finally decrypt with K_1.

Due to this design of Triple DES as an encrypt–decrypt–encrypt process, it is possible to use a 3TDES (hardware) implementation for single DES by setting $K_{1,}$ $K_{2,}$ and K_3 to be the same value. This provides backwards compatibility with DES.

Second variant of Triple DES (2TDES) is identical to 3TDES except that $K_3$is replaced by K_1. In other words, user encrypt plaintext blocks with key $K_{1,}$ then decrypt with key $K_{2,}$ and finally encrypt with K_1 again. Therefore, 2TDES has a key length of 112 bits.

Triple DES systems are significantly more secure than single DES, but these are clearly a much slower process than encryption using single DES.