**DES ALGORITHM**

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 algorithm design 64 bits plaintext used as input. The algorithm transforms input into series of block which is 64 bits cipher text.16 rounds of encryption process is handled for every plaintext block. Decryption process is done in reverse manner of encryption method, by introducing sulky ki introduced by main key k where i=1……….16.

***Specification***

Designers IBM

First published 1975 (Federal Register) (standardized in January 1977)

Successors Triple DES, G-DES, DES-X, LOKI89, ICE

Key sizes 56 bits (+8 parity bits)

Block sizes 64 bits

Structure Balanced Feistel network

Rounds 16

***DES Explanation***

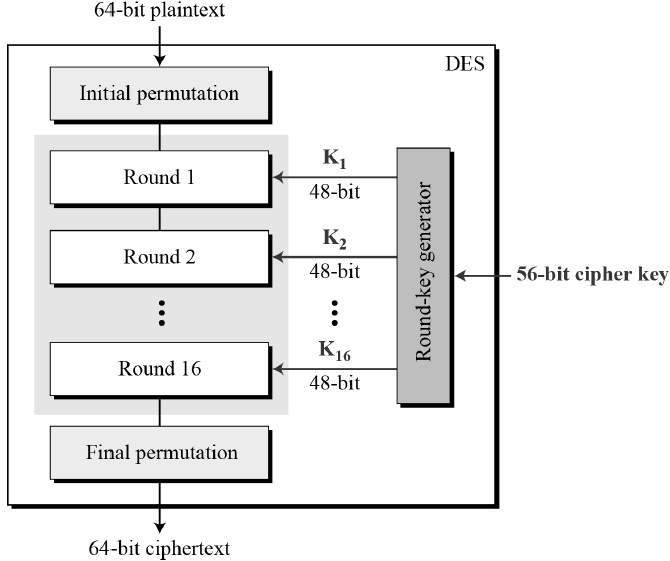
Expansion:

The 32-bit half-block is expanded to 48 bits using the expansion permutation, denoted E in the diagram, by duplicating half of the bits. The output consists of eight 6-bit (8 \* 6 = 48 bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side.

Key mixing: The result is combined with a subkey using an XOR operation. Sixteen 48-bit subkeys—one for each round—are derived from the main key using the key schedule (described below).

Substitution:

After mixing in the subkey, the block is divided into eight 6-bit pieces before processing by the S-boxes, or substitution boxes. Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table. The S-boxes provide the core of the security of DES—without them, the cipher would be linear, and trivially breakable.

Fig.3. Data Encryption Standard Algorithm Procedure

Permutation:

Finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the P-box. This is designed so that, after permutation, the bits from the output of each S-box in this round are spread across four different S-boxes in the next round.

# Algorithm to Encrypt an Image using DES Algorithm

1: Select an Image to Encrypt.

2: Declare a dynamic binary array of size = height of image \* width of image \* RGB\_size.

The RGB\_ size is 3.

3: Calculate Binary Value of each pixel of an Image.

4: Store all the RGB values of the pixel into the array.

5. Divide the binary array into blocks of size 64 bits.

6. This bits is split into two 32-bit halves designated L(left) and R(right).

7.At the same time, the first subkeys K1, a 48-bit string is generated.

8. The subkey K1 analog with the right halve R are used as inputs to the round function F(K;R(x)) to produce a 32-bit output, blow we explain briefly the steps of the round function F:

* Expand x from 32 bits to 48-bit, by using the expansion box E, see Fig. 3(a).
* Apply the modulo 2 addition of E(x) and K, the output is also 48-bit.
* Where the later is concatenation of eight bit string Bi of length six, say . Enter each Bi into S-box where S-box is generated from a linear function, witch takes six bits as an inputs and get four outputs.
* The output of the pervious step has a 32-bit length is entered into the permutation function P, which is defined as P box.

9. The output from the round function F is XOR-ed with the left half of the plaintext.

10. Finally, the left old half of the plaintext is replaced by the old right half, and the output of the XOR replaces the old value of R.

11. This completes one round of the DES. The same procedure is applied 15 more times, the only difference being the subkeys K2, K3,...,K15 generated by the subkey schedule are used as inputs to the round function f.

12. The last step of encryption is to reassemble the L and R output by the last round of 64-bit string and apply the inverse of initial permutation . This is the Cipher bit array.

13. Declare a dynamic binary array of size = height of image \* width of image \* RGB\_size. Now arrange the cipher values into the image .

14. The image now formed is the encrypted image file

15. Exit.