Name : Ansari M.Saeem M.Saleem

Uid : 2019430001

Subject : NAD

Expt no : 7

Aim : Write a program for encrypting 64 bit playing text using DES algorithm.

Aim: Write a program for encrypting 64 bit playing text using DES algorithm.

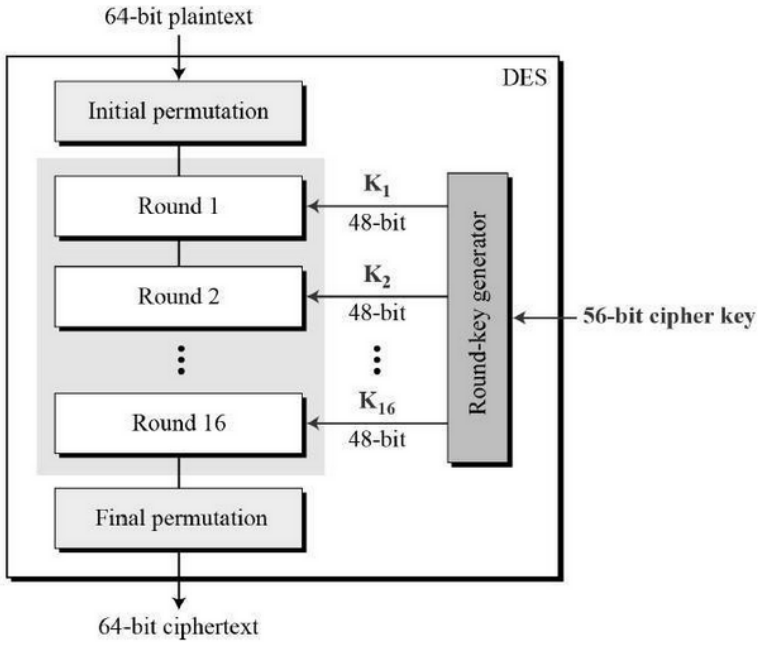
Objectives:

- To encrypt and decrypt plain text using DES algorithm in order to provides secure communication over the network.
- To define the basic structure and working of DES

Theory:

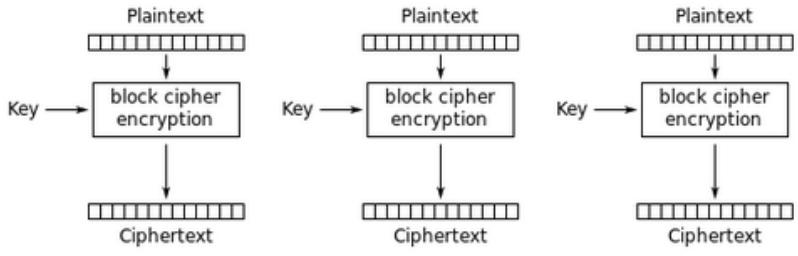
- The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).
- DES uses 16 round structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only)

General Structure of DES:



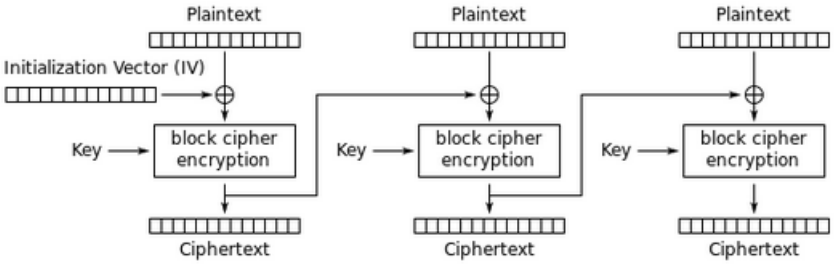
Modes of operation :

ECB : If each 64 bit is encrypted or decrypted independently, then this mode is ECB.

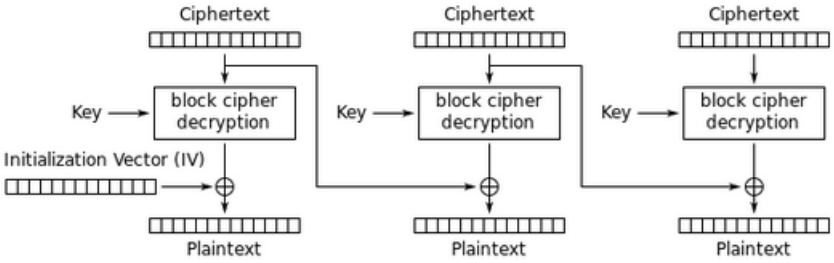


Electronic Codebook (ECB) mode encryption

CBC: If each 64-bit data is dependent on the previous one, then this mode is called CBC or CFB mode. Here Initial Vector (IV) is same as the data block size.



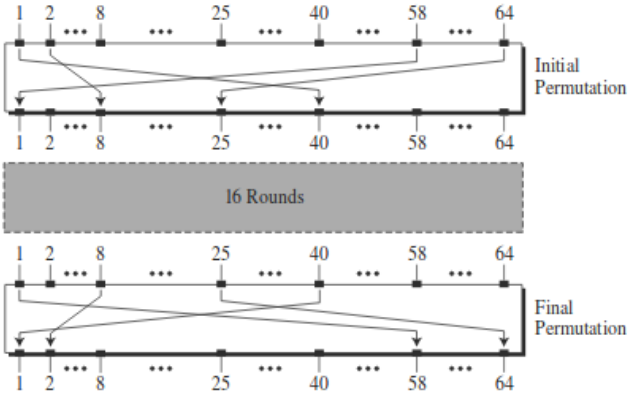
Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Methodology :

Initial and final permutation : Each of these permutations takes a 64-bit input and permutes them according to a predefi ned rule.These permutations are keyless straight permutations that are the inverse of each other.For example, in the initial permutation, the 58th bit in the input becomes the fi rst bit in the output. Similarly, in the fi nal permutation, the fi rst bit in the input becomes the 58th bit in the output.

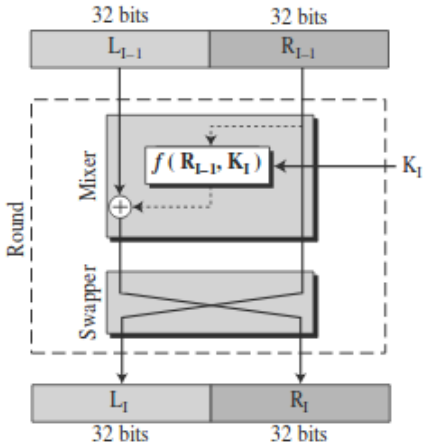


Initial and final permuatation

Initial Permutation	*Final Permutation*
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

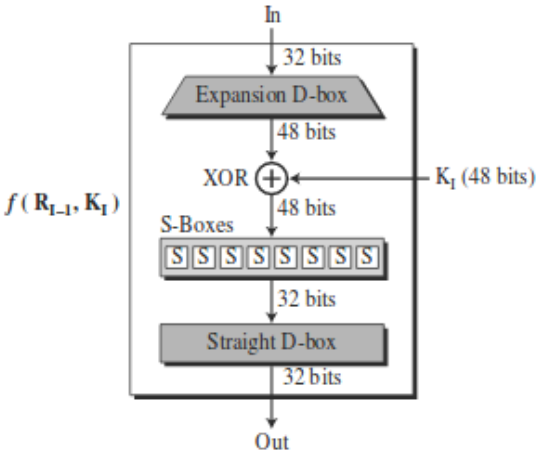
Initial & final permutation table

Rounds : DES uses 16 rounds. Each round of DES is shown in Fig. The round takes LI−1 and RI−1 from previous round (or the initial permutation box) and creates LI and RI, which go to the next round (or final permutation box).We can assume that each round has two cipher elements (mixer and swapper). The Swapper swaps the left half of the text with the right half. The mixer is invertible because of the XOR operation. All non-invertible elements are collected inside the functionf (RI−1, KI).



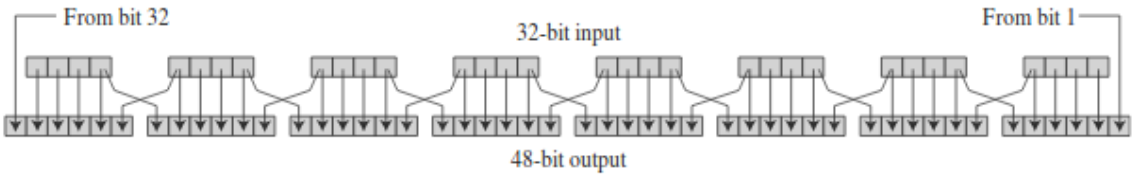
Rounds in DES

DES Function : The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits (RI−1) to produce a 32-bit output. This function is made up of four sections: an expansion D-box, a whitener (that adds key), a group of S-boxes, and a straight D-box.



DES Function

Expansion D-box : Since RI−1 is a 32-bit input and KI is a 48-bit key, we fi rst need to expand RI−1 to 48 bits. RI−1 is divided into 8 4-bit sections. Each 4-bit section is then expanded to 6 bits. This expansion permutation follows a predetermined rule. For each section, input bits 1, 2, 3, and 4 are copied to output bits 2, 3, 4, and 5, respectively. Output bit 1 comes from bit 4 of the previous section; output bit 6 comes from bit 1 of the next section. If sections 1 and 8 can be considered adjacent sections, the same rule ap-plies to bits 1 and 32



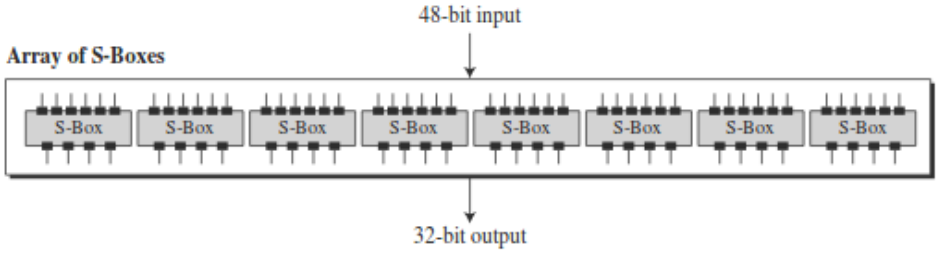
Expansion Permutation

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

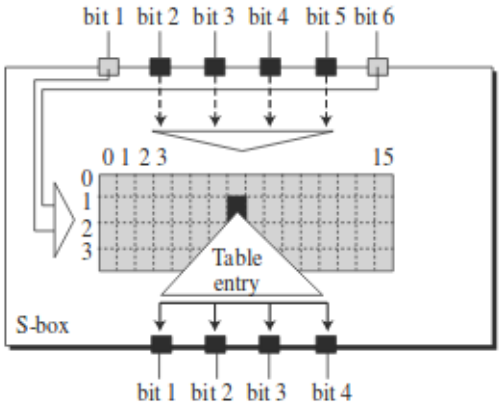
Expansion Table

XOR : After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

S-Boxes : The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.



The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box. The result of each box is a 4-bit chunk; when these are combined the result is a 32-bit text. The substitution in each box follows a pre-determined rule based on a 4-row by 16-column table. The combination of bits 1 and 6 of the input defi nes one of four rows; the combination of bits 2 through 5 defi nes one of the sixteen columns as shown in Fig. 6.8. This will become clear in the examples. Because each S-box has its own table, we need eight tables, as shown in Tables 6.3 to 6.10, to defi ne the output of these boxes. The values of the inputs (row number and column number) and the values of the outputs are given as decimal numbers to save space. These need to be changed to binary



Results :

```
students@CE-Lab3-603-U22:~/Desktop/saeem/NAD/7$ python des2.py
Round 0 : ©[]Ã³
Round 1 :@# A^'
Round 2 :.a[]í"
Round 3 :3Ms[][]
Round 4 :aDw O[]
Round 5 :_
          _[]
Round 6 :eí[][]¡§
Round 7 :U[][]
Round 8 :[][] µÇÖ
Round 9 :X»a®[]
Round 10 :E[][]¼e
Round 11 :õÒUóB[]
Round 12 :¡ ýP%
¤ð@nd 13 :
:
Round 14 :A»[][]X
Round 15 :/×¡Ì
             í
Ciphered: 'ï¤30È´\x14É'
Deciphered:  Hello
```

Conclusion:

Here we can conclude that DES algorithm can be used to encrypt and decrypt plain text in order to provide secure communication in the network.