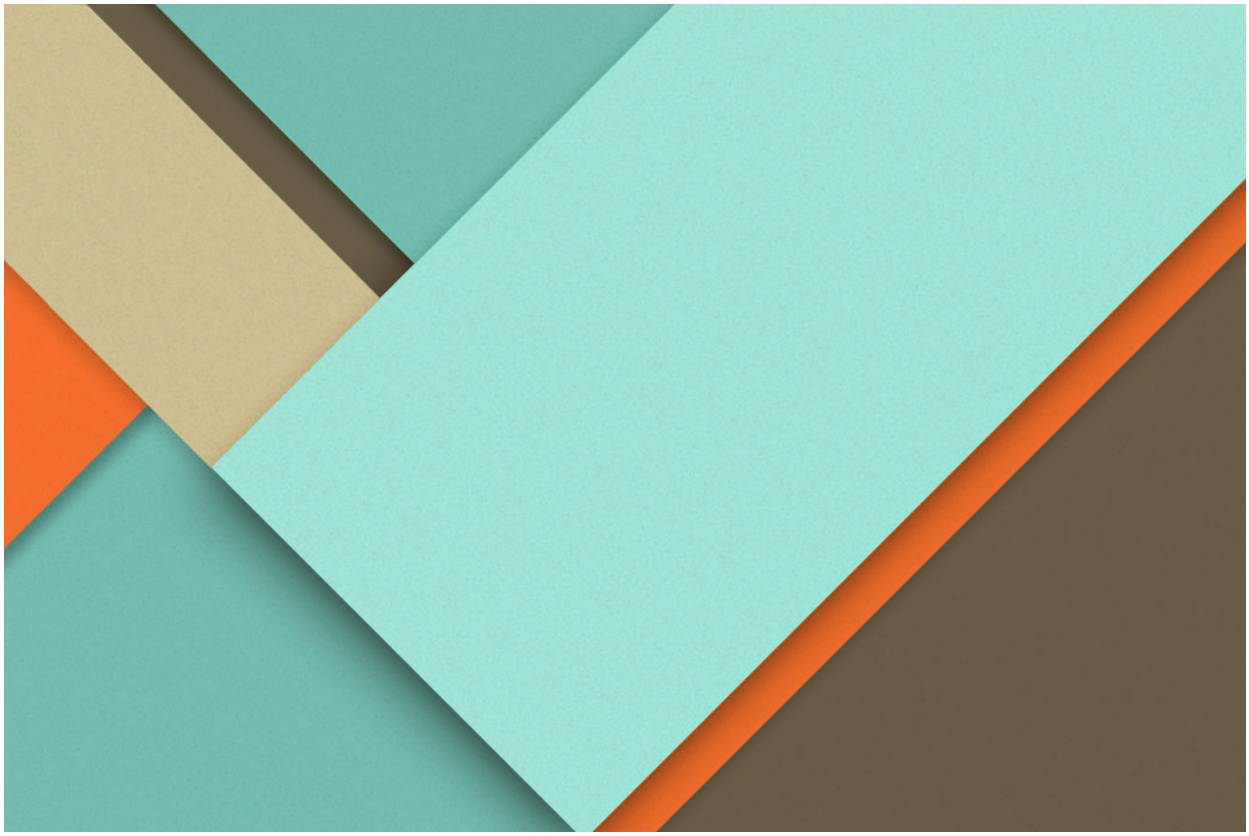

DES (Data Encryption Standard)

—

Parveen 2021079

Shubham Sharma 2021099

Description

This Assignment Explores the Implementation of DES (*Data Encryption Standard*) and the Verification of Implementation by matching the Deciphered text with the Original Plain text.

I. It is a symmetric key algorithm used for Encryption and Decryption.

II. It uses multiple sub-key generation rounds & a Feistel cipher structure.

III. For decryption, the process is essentially the reverse of encryption, using the subkeys in the **reverse** order.

IV. Also, DES produces a strong Avalanche effect; that is, a change in one bit of input produces a change in many bits of output.

V. Later on, it was considered insecure due to its small key size (56 bits).

Constraints & Details

Input block size: 64-bit blocks

Initial key length: 64-bit

Key length after parity drop: 56-bit

Sub key length: 48-bit

No. of rounds: 16

Output block size: 64 bit

Uses **FEISTEL** cipher like structure

Key Components

I. Initial permutation

A. Rearranges the bits of the input plaintext.

B. Provides diffusion in the Encryption process.

II. Expansion Permutation Box

A. Increases the amount of data to be processed by each S-box.

B. Provides more diffusion in the encryption process.

III. S-boxes (Substitution Boxes)

A. Non-linear components that substitute groups of bits from the input.
B. Introduce non-linearity and confusion into the algorithm.

IV. P-box (Permutation Box)

A. Rearranges the bits after the S-box substitution.
B. Further enhances the diffusion of the algorithm.

V. Parity Drop Box

A. Drops parity bits from the key.
B. Reduces the key size from 64 to 56 bits.

VI. Final Permutation Box:

A. Reorders the bits in the final output ciphertext.
B. Provides a reversible transformation for decryption.

Encryption & Decryption

Encryption

I. STEP 1: Key Generation

A. Generate round keys by shifting halves of the key.

II. STEP 2: Encryption Rounds

A. Apply initial permutation on plain text.
B. Perform expansion permutation and XOR with the key.
C. Substitute using S-boxes.
D. Apply the permutation box and XOR with the left half.
E. Repeat for 16 rounds.

III. STEP 3: Final Permutation

A. Apply the final permutation to obtain the Cipher text.

Decryption

I. STEP 1: Key Generation

A. Generate round keys by shifting halves of the key.

II. STEP 2: Decryption Rounds

A. Apply initial permutation to Cipher Text.
B. Perform expansion permutation and XOR with the keys in reverse order.
C. Substitute using S-boxes.
D. Apply the permutation box and XOR with the left half.
E. Repeat for 16 rounds.

III. STEP 3: Final Permutation

A. Apply the final permutation to obtain the Plain Text.

```
def encrypt(left, right, key):
    '''Apply Expansion Permutation and XOR with Key'''

    expansion_result = "".join([right[i - 1]
    |   |   |   |   |   |   |   for i in expansion_permutation_Box])
    xor_result = str(bin(int(expansion_result, 2) ^ int(key, 2)))[2:].zfill(48)

    '''Apply S-Box Substitution'''

    S_Str = ""
    for i in range(0, 8):
    |   S_row = xor_result[(i * 6)] + xor_result[(i * 6) + 5]
    |   S_col = xor_result[(i * 6) + 1] + xor_result[(i * 6) + 2] + \
    |   |   xor_result[(i * 6) + 3] + xor_result[(i * 6) + 4]
    |
    |   S_Str += dec_to_bin(S_Box[i][bin_to_dec(S_row)][bin_to_dec(S_col)])

    '''Apply Permutation Box and XOR with Left Half'''

    return right, str(bin(int(left, 2) ^ int("".join([S_Str[i - 1] for i in P_BOX]), 2)))
```

The function used for Encryption & Decryption is the same.

Results

I. Comparing the final decipher text with the original plain text verified that the code works as per the DES algorithm.
II. Verifies the Output of Encryption Round 1 is SAME as the Output of Decryption

Round 15.

III. Verifies the Output of Encryption Round 14 is SAME as the Output of Decryption Round 2.

```
For Plain Text 1:-

Output of Encryption Round 1  is 0000000010010101000010011000000111000010100101001101111001010000
Output of Decryption Round 15 is 0000000010010101000010011000000111000010100101001101111001010000

Output of Encryption Round 1 is SAME as Output of Decryption Round 15


Output of Encryption Round 14  is 0101100011001111001011001000110001011011100011001100101000010101
Output of Decryption Round 2 is 0101100011001111001011001000110001011011100011001100101000010101

Output of Encryption Round 14 is SAME as Output of Decryption Round 2


Encryption and Decryption SUCCESSFUL
```

Key and Sample Input Output

Key: $hk5w^N+

I. Input (Plain Text)

+C$#W^f5j?v3EQVHA{qA%WS(w.&&?4aUUrK.r3Yc*6vmmx,taU6wzRaFf[rFLfK5Q1Qe$g_=?@dH1PPv

Cipher Text

101110001000000110101010001001110111111110010110011100000111001111101111001001001100100100110000111110011000101010101100110111000000010011111001111000011111010101001000111101001101000111010111100110001001110011111111110001110100010001100110101010000110010010110111001011010100011010010110001111001101100101111111001000111001000001111101000111010000101101011000101010001011101000100110010100101100010011100110111010101000000001100110111111100011110011000010101010101111011100011001001001110100000110011100001101111010100110000101111101100010011110000001010000110110100000010010111101100100001010100011110100111110011001011010010110001011000011

II. Input (Plain Text)

ReBtyD,!]PaHcn6Gy0_{./YA,6kLRCy?(&/*WrbdGUf,]SfVR&tjhTpap6N]HkrWgXU@?Y{2f?{0fqGQ

Cipher Text

01110111001110011011100001000010100101110001101010010000010000110010
01101001101010001111010101001011111001011110101111010110001111001110
10111000011101110100011011011111110001011111001001010100001110000101
00100111110001001011101100100110001111100111010110000111010100100001
10100110101011000110101111010110010111110010100000001100011100011100
01011000101001111011110101111011000000100101011000001001010101000000
01011111111110010101000011110100111110111101010111010111010001001111
10010111111101110100110101110111001000100000010010101011111010100001
01100111111111001011101001111000111100110100100000010110101001100111
11011111110000110101110101110

III. Input (Plain Text)

64XvR]L.Q+yck:05N]8MH]XAtbEXvFJ+T2&@N$kRJ0wCZBC?B{d+;Tnu,.+(uM6rY*)tuFnezQyRdD-G

Cipher Text

11001110101101010101000101000010011110001110001110100110101010000010
10111011101011011100110001010001010110011001100110000010100000010110
10100000101101110010010100011011010111000101110111001110010110011100
11111001111100101111010100100111011111110111110111010001001101111010
01101000111110111001111110011000011010000100100000110001000110000011 0
01011111001111110000001100000010111111000111101111000100001000001101
11100000100001001001000010100000000111110010110100010010101010100000
00111101100100110001011000100100000011110111001010100100000000011111
10010001000011001000001101001101000111010111001011010001111001110111
10100110110000001111111011100

IV. Input (Plain Text)

#z=gV?[(wtX[SNjhB=(J{ZKYBg,!1]dz@c2{qNvb1:V0]T87ZSpySa&.d/%6_nGxMPgaf;{_+(-C)b@i

Cipher Text

0001001000000110001011110100101000001111010110011101101010000001001 1
1010001110110101100000110000111101110100101101100010101101001110010 0
0111011100111011011000000000010010000100000010001101101110101110011 0
1101000100110010010101001110000110111110000111101000111001100110010 1
1010110111101010001110010000110010101101111101010110101110010010000 1
0000101010001001010010000011000011110100101100110101000111110010101 0
0000101011001101000101110101000010111110011110100111110000000010110 0
1010111101111101111110001110100101111001101001010001011101111011101 1
1001001111100111100101010111110101010011010001110101110000111011111 1
0011101111111101000110101011

V. Input (Plain Text)

&eqw+Cu=QBkA_+Eb4[T$iR9WM$cd2vzEB_mC8-*S+cn((KwkWLu]2v:GY03iUUFru]jp_9
:;y0{xFi)!

Cipher Text

1000111110100111100001000000000111101000100011000111000110011010000 0
1100010000101101011101000011000000001110010011101011101110101100100 1
1000111001100000101100111101010010010000100100100011110011111010110 1
1000111001110001001111011100001010000100100111110010010000000100011 10
1101010110110010110111000101101011000000110010110000011010000101111 0
1101010001001111001001100111010010011011010111111011110111110101011 1
0101000011100001100010011010000010111001101010110101101011100001001 1
1000111100100110110001100101110101010011000001010001001010101010101 0
0000101011110101100110000010011000110001100001000011100000100110000 1
1000111011100000100011110111