Kartik Padave

A022

70362019039

Experiment 4

Aim

To implement DES.

Theory

DES is a block cipher and encrypts data in blocks of size of 64 bits each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits.

We have noted initial 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the initial key. Thus, for each a 56-bit key is available. From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called key transformation. For this, the 56-bit key is divided into two halves, each of 28 bits. These halves are circularly shifted left by one or two positions, depending on the round. After an appropriate shift, 48 of the 56 bits are selected. for selecting 48 of the 56 bits the table is shown in the figure given below. For instance, after the shift, bit number 14 moves to the first position, bit number 17 moves to the second position, and so on. If we observe the table carefully, we will realize that it contains only 48-bit positions. Bit number 18 is discarded (we will not find it in the table), like 7 others, to reduce a 56-bit key to a 48-bit key. Since the key transformation process involves permutation as well as a selection of a 48-bit subset of the original 56-bit key it is called Compression Permutation. Because of this compression permutation technique, a different subset of key bits is used in each round. That makes DES not easy to crack.

After the initial permutation, we had two 32-bit plain text areas called Left Plain Text(LPT) and Right Plain Text(RPT). During the expansion permutation, the RPT is expanded from 32 bits to 48 bits. Bits are permuted as well hence called expansion permutation. This happens as the 32-bit RPT is divided into 8 blocks, with each block consisting of 4 bits. Then, each 4-bit block of the

previous step is then expanded to a corresponding 6-bit block, i.e., per 4-bit block, 2 more bits are added. This process results in expansion as well as a permutation of the input bit while creating output. The key transformation process compresses the 56-bit key to 48 bits. Then the expansion permutation process expands the 32-bit RPT to 48-bits. Now the 48-bit key is XOR with 48-bit RPT and the resulting output is given to the next step, which is the S-Box substitution.

Code

```
from des import DesKey

key = DesKey(b"AABB09182736CCDD")
print(key)

plain_text = "My name is Kartik"
print(f"Plain Text: {plain_text}")

encrypt_msg = key.encrypt(b"My name is Kartik", padding=True)
print(f"Encrypted Message: {encrypt_msg}")

decrypt_msg = key.decrypt(encrypt_msg, padding=True)
print(f"Decrypted Message: {decrypt_msg}")
```

Output

```
E:\Programs\College-Labs\CRYPTO-Lab\ProgFiles>python DES.py
<des.base.DesKey object at 0x00000230315CB2E0>
Plain Text: My name is Kartik
Encrypted Message: b'\xcd\xc17K\x00}\xcc\xe4\xc4\t"#U\xaeK@\xa3\x8a\xae\x0c\xc1\xc5\xe3\xb4'
Decrypted Message: b'My name is Kartik'
```

Conclusion

Hence, we were able to implement DES.