

Roll. No. A016	Name: Varun Khadayate
Class B.Tech CsBs	Batch: 1
Date of Experiment: 27-08-2021	Subject: Cryptology

Aim

To Implement DES Encryption and Decryption standard

Theory

Until the introduction of the Advanced Encryption Standard (AES) in 2001, the Data Encryption Standard (DES) was the most widely used encryption scheme. DES was issued in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). The algorithm itself is referred to as the Data Encryption Algorithm (DEA).⁷ For DEA, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.

Over the years, DES became the dominant symmetric encryption algorithm, especially in financial applications. In 1994, NIST reaffirmed DES for federal use for another five years; NIST recommended the use of DES for applications other than the protection of classified information. In 1999, NIST issued a new version of its standard (FIPS PUB 46-3) that indicated that DES should be used only for legacy systems and that triple DES (which in essence involves repeating the DES algorithm three times on the plaintext using two or three different keys to produce the ciphertext) be used. We study triple DES in Chapter 6. Because the underlying encryption and decryption algorithms are the same for DES and triple DES, it remains important to understand the DES cipher.

The overall scheme for DES encryption is illustrated in Figure 3.5. As with any encryption scheme, there are two inputs to the encryption function: the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.⁸

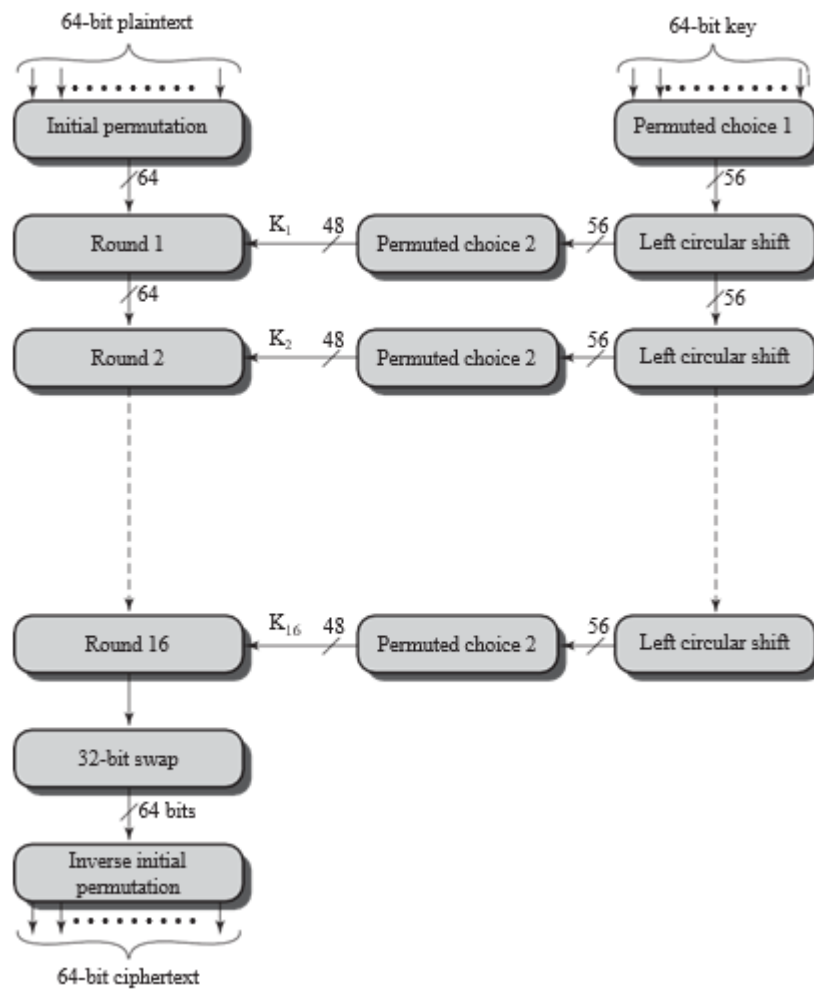
Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input*. This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the **preoutput**.

Finally, the preoutput is passed through a permutation $[IP^{-1}]$ that is the inverse of the initial permutation function, to produce the 64-bit ciphertext. With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher, as shown in Figure 3.3.

The right-hand portion of Figure 3.5 shows the way in which the 56-bit key is used. Initially, the key is passed through a permutation function. Then, for each of the sixteen rounds, a *subkey* (K_i) is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

DES Decryption

As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of the subkeys is reversed. Additionally, the initial and final permutations are reversed.



Code

```
from pyDes import *

data = "Hello my name is Varun Khadayate"
k = des("DESCRYPT", CBC, "\0\0\0\0\0\0\0\0", pad=None, padmode=PAD_PKCS5)
d = k.encrypt(data)

while True:
    ch = int(input('Welcome to DES Encryption and Decryption Program Made by  
Varun Khadayate..\n [*] Press 1 for Encryption \n [*] Press 2 for Decryption  
\n [*] Press 0 to exit..\n \nYour Choice:: '))

    if ch == 1:
        print("\n=====")
        print("                !!!!Encryption!!!!                ")
        print("Enter the Plain Text: ",data)
        print("Encrypted Text is : ", d)
        print("\n=====\\n\\n")

    elif ch == 2:
        print("\n=====")
        print("                !!!Decryption!!!                ")
        print("Enter the Encrypted Text: %r", d)
        print("Decrypted Text is: ",k.decrypt(d))
        print("\n=====\\n\\n")

    elif ch == 0:
        print("\n=====")
        print("        Thank You for using the Software ;)        ")
        print("                Exiting Now.                ")
        print("=====\\n\\n")
        exit()
```

Output

```
PS E:\College-Codes\Fourth Year\SEM VII> & C:/Users/varun/AppData/Local/Programs/Python/Python310/python.exe "e:/College-Codes/Fourth Year/SEM VII/CT/DES.py"
Welcome to DES Encryption and Decryption Program Made by Varun Khadayate..
[*] Press 1 for Encryption
[*] Press 2 for Decryption
[*] Press 0 to exit..

Your Choice:: 1

=====
!!!Encryption!!!
Enter the Plain Text: Hello my name is Varun Khadayate
Encrypted Text is : b'!\x02>g96\x10$~\xbd\xc0?5\x83\xf9\x01\xd7\t\xbdYn\xe9|l\x9a\xef\x8f\xa6\x8aN5\xd9U\x97H\x1d\xb4N\x06\xd7'

=====

Welcome to DES Encryption and Decryption Program Made by Varun Khadayate..
[*] Press 1 for Encryption
[*] Press 2 for Decryption
[*] Press 0 to exit..

Your Choice:: 2

=====
!!!Decryption!!!
Enter the Encrypted Text: %r b'!\x02>g96\x10$~\xbd\xc0?5\x83\xf9\x01\xd7\t\xbdYn\xe9|l\x9a\xef\x8f\xa6\x8aN5\xd9U\x97H\x1d\xb4N\x06\xd7'
Decrypted Text is: b'Hello my name is Varun Khadayate'

=====

Welcome to DES Encryption and Decryption Program Made by Varun Khadayate..
[*] Press 1 for Encryption
[*] Press 2 for Decryption
[*] Press 0 to exit..

Your Choice:: 0

=====
Thank You for using the Software ;)
Exiting Now.
=====
```

Conclusion

Hence, we were able to perform DES encryption and Decryption.