Walchand College of Engineering, Sangli (Government Aided Autonomous Institute)	
AY 2025-26	
Course Information	
Programme	B.Tech. (Computer Science and Engineering)
Class, Semester	Final Year B. Tech., Sem VII
Course Code	6CS451 C
Course Name	Cryptography and Network Security Lab
PRN	22510016

Experiment No. 05

Title - ApplyDES algorithm for practical applications

Objectives:

To implement the DES (Data Encryption Standard) algorithm for encrypting and decrypting a plaintext message to ensure secure data transmission. This lab demonstrates how symmetric key cryptography can be used to protect information in real-world communication systems.

Problem Statement:

You are working as a software security analyst for a company that needs to transmit sensitive employee data (like Social Security Numbers or salary information) between two systems over a network. To ensure data confidentiality, you are required to implement the ***DES algorithm*** in encryption and decryption mode.

Your task is to write a program that:

1. Takes a plaintext input from the user.
2. Uses a user-defined 64-bit key (as a hexadecimal or ASCII string).
3. Encrypts the plaintext using the DES algorithm.
4. Outputs the encrypted text (ciphertext).
5. Decrypts the ciphertext back to plaintext using the same key to verify correctness.

Equipment/Tools:

Computer or laptop

Python 3.x

PyCryptodome library (`pip install pycryptodome`)

Code editor/IDE (VS Code.)

Theory:

DES is a symmetric-key block cipher operating on 64-bit blocks with an effective 56-bit key (8 parity bits → 64 bits total). It uses a 16-round Feistel structure with S-boxes and permutations. In practice, DES is used in a block mode (e.g., CBC). CBC mode uses an Initialization Vector (IV) to randomize encryption of identical blocks. PKCS#7 padding is used when plaintext length is not a multiple of 8 bytes.

Procedure:

1. Collect plaintext from the user.
2. Read a 64-bit key from the user as either:
   16 hex characters (e.g., 133457799BBCDFF1), or
   8 ASCII characters (e.g., mykey123).
3. Validate the key length and convert it to 8 bytes.
4. Encrypt using DES in CBC mode with a fresh random 8-byte IV; apply PKCS#7 padding.
5. Display the IV and ciphertext (both in hex).
6. Decrypt using the same key and IV; remove padding.
7. Verify that decrypted text equals the original plaintext.

Steps:

`Install library: pip install pycryptodome`

Save and run the Python code below.

Enter plaintext and key when prompted.

Observe ciphertext and the recovered plaintext.

Program code:

```
fromcryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
fromcryptography.hazmat.backends import default_backend

def pad(text):
    whilelen(text)%   8!= 0:
        text += ' '
    return text

defencrypt_3DES(key,  plaintext):
    plaintext =pad(plaintext).encode('utf-8')
    cipher=  Cipher(algorithms.TripleDES(key.encode('utf-8')), modes.ECB(),
backend=default_backend())
    encryptor =cipher.encryptor()
    ciphertext=encryptor.update(plaintext) + encryptor.finalize()
    returnciphertext.hex().upper()

defdecrypt_3DES(key,  ciphertext_hex):
    ciphertext=bytes.fromhex(ciphertext_hex)
    cipher=  Cipher(algorithms.TripleDES(key.encode('utf-8')), modes.ECB(),
backend=default_backend())
    decryptor =cipher.decryptor()
    decrypted =decryptor.update(ciphertext) + decryptor.finalize()
    returndecrypted.decode('utf-8').rstrip()


key="12345678abcdefgh12345678"    # 24 bytes for TripleDES

texts=["HelloWorld","DES Algorithm", "SecureData123", "ConfidentialMsg",
"SalaryInfo2025"]

for text in texts:
    enc=encrypt_3DES(key,     text)
    dec=decrypt_3DES(key,      enc)
    print("Plaintext:",text)
    print("Encrypted:",enc)
    print("Decrypted:",dec)
    print()
```

Output

```
PS C:\Users\aishw\OneDrive\Documents\Final Year\CNS Practicals\Assignment_5_23520001> python des_algo.py
Plaintext: HelloWorld
Encrypted: 2ECE85A420B0234C95CB6581C90E5B3B
Decrypted: HelloWorld

Plaintext: DES Algorithm
Encrypted: E580FA9328CDB349793B42F9E5594B09
Decrypted: DES Algorithm

Plaintext: SecureData123
Encrypted: 02A47F4CC6BB28420784F522616F583B
Decrypted: SecureData123

Plaintext: ConfidentialMsg
Encrypted: 9FE129B6B8159837CD7128E57B362D3E
Decrypted: ConfidentialMsg

Plaintext: SalaryInfo2025
Encrypted: 3CCD266A712D42897A9AAADF4B032429
Decrypted: SalaryInfo2025
```

Observations and Conclusion:

DESsuccessfully encrypts and decrypts the given plaintext using the same key.

Theciphertext is unreadable to an unauthorized party, ensuring data confidentiality.

Thedecrypted output matches the original plaintext, proving correctness of the implementation.

Thisexperiment demonstrates the use of ***symmetric cryptography*** for secure communication in practical applications.