LAB 3

Decryption

Step 1: Install Required Library

Step 2 : making sure we have the key( key\_iv.txt) for decryption and also the encrypted file.

Step 3: Decryption Program

* Reads the encrypted file (encrypted.txt).
* Decrypts the data using AES with the saved key and IV.
* Strips trailing spaces that were added during encryption.
* Writes the decrypted content to decrypted.txt.

Python program

from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes

from cryptography.hazmat.backends import default\_backend

def decrypt\_file(encrypted\_file, output\_file, key, iv):

with open(encrypted\_file, 'rb') as file:

encrypted\_data = file.read()

cipher = Cipher(algorithms.AES(key), modes.CFB(iv), backend=default\_backend())

decryptor = cipher.decryptor()

decrypted\_data = decryptor.update(encrypted\_data) + decryptor.finalize()

# Remove any trailing spaces added during encryption

decrypted\_data = decrypted\_data.rstrip(b' ')

with open(output\_file, 'w') as file:

file.write(decrypted\_data.decode('utf-8'))

# Read the saved key and IV from file

with open('key\_iv.txt', 'rb') as file:

key\_iv = file.read()

key = key\_iv[:32] # First 32 bytes are the AES key

iv = key\_iv[32:] # Last 16 bytes are the IV

# Decrypt the file

decrypt\_file('encrypted.txt', 'decrypted.txt', key, iv)

print("Decryption complete! Decrypted data written to 'decrypted.txt'")