Name: Madhuri Ramakrishnan

119A3032

TE-IT(E1)

EXPERIMENT-04

Aim: Encrypt long messages using various modes of operation using AES or DES.

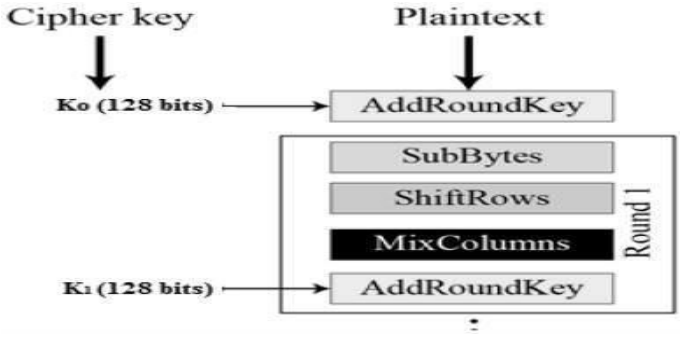
Resources Used:

1. VS Code IDE
2. Programming Language – Python

Theory:

- AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'.
- It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).
- AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes.
- These 16 bytes are arranged in four columns and four rows for processing as a matrix.
- Unlike DES, the number of rounds in AES is variable and depends on the length of the key.
- AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.

Encryption Process



Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row.

MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes.

Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key.

All the steps are performed for 10, 12 or 14 rounds, depending upon the plaintext.
MixColumns is not performed in the last round.

Procedure/Algorithm

- The first thing we are going to do is importing the **AES** module from the pycrypto library.
- Next we need to set our secret encryption key.
- The length of the key needs to be **16, 24 or 32 bytes long**, depending if we want to use **AES-128, AES-192 or AES-256** respectively
- We are going to choose an arbitrary **16** bytes key just for illustrations purposes.
- Next we need to call the **new** function of the **AES** module. This function will return an object of class AESCipher , which provides the functions to both encrypt and decrypt the data.

- The **key** parameter corresponds to the encryption key to be used by the algorithm and we will pass it the key we previously defined.The **mode** parameter corresponds to the chaining mode that is used for decryption / encryption. We are going to pass the value **MODE_ECB**, to use the electronic code book mode.
- Now that we have our **AESCipher** object, we can encrypt the data with a call to the **encrypt** method.
- This **encrypt** method call will return as output a string with the cipher text.
- Now, we will convert the cipher text to its hexadecimal representation. To do it, we call the **encode** method on our cipher text string, passing the value **"hex"** as input.
- Now that we have our cipher text, we will decrypt it back to plain text.
- Finally, we call the **decrypt** method on our new object, passing as input the ciphered text. It returns as output the original decrypted plain text, which we will print.

Code:

```
from Crypto.Cipher import AES

key = 'abcdefghijklmnop'

cipher = AES.new(key, AES.MODE_ECB)
msg =cipher.encrypt('hello sies world')

print("Encrypted: ",msg.encode("hex"))

decipher = AES.new(key, AES.MODE_ECB)
print("Decrypted: ",decipher.decrypt(msg))
```

Results:

```
Success #stdin #stdout 0.01s 7440KB
('Encrypted: ', 'ef4c26587bb063f41e408b7ac8fa1982')
('Decrypted: ', 'hello sies world')
```

Conclusion:

In this experiment we have learned about:

- ✓ Working of Advanced Encryption Standard (AES)
- ✓ Implemented encryption and decryption of plain text keeping the key as 'abcdefghijklmnop' and verified the results.

Hence, we successfully encrypted and decrypted a long message using AES method.