

Assignment No: 2

Title: Implement AES Algorithm

Name: Shubhankar jakate

PRN : 22310371

Roll No = 382019

Objective: The key objectives of this assignment are:

1. To study the concept of symmetric key cryptography.
2. To understand the Advanced Encryption Standard (AES) algorithm.
3. To implement AES for text encryption and decryption using Python.

Theory:

Introduction to Symmetric Cryptography

In symmetric key cryptography, the same secret key is used for both encryption and decryption. This makes it faster and more efficient than asymmetric key cryptography. However, the challenge is secure key distribution between sender and receiver.

Advanced Encryption Standard (AES)

AES is the most widely used symmetric encryption algorithm in the world today. It was adopted as a standard by the **National Institute of Standards and Technology (NIST)** in **2001**, replacing the older DES (Data Encryption Standard).

- **Type:** Block Cipher
- **Block Size:** 128 bits (16 bytes) fixed
- **Key Size:** 128, 192, or 256 bits
- **Rounds:**
 - 10 rounds for 128-bit key
 - 12 rounds for 192-bit key
 - 14 rounds for 256-bit key

AES Structure

AES works on a 4×4 matrix of bytes, called the State. Each round of AES encryption applies a sequence of transformations to this matrix.

Main Steps in AES Encryption:

1. **Key Expansion**
 - The original key is expanded into multiple round keys using a process called Rijndael Key Schedule.
 - Each round uses a different key derived from the original secret key.
2. **Initial Round (Pre-round transformation):**

- AddRoundKey: The state matrix is XORed with the first round key.
- 3. Main Rounds (9, 11, or 13 depending on key size):**
- SubBytes: Each byte is substituted with another byte from the S-box (Substitution box). This provides non-linearity to the cipher.
 - ShiftRows: The rows of the state are shifted cyclically by different offsets. This provides diffusion.
 - MixColumns: Each column is mixed using mathematical operations in Galois Field ($GF 2^8$). This step further spreads the influence of each byte across the column.
 - AddRoundKey: The state is XORed with the round key.
- 4. Final Round (Without MixColumns):**
- SubBytes → ShiftRows → AddRoundKey

AES Decryption

The decryption process follows the same structure but in reverse order. Instead of using S-Box, it uses **Inverse S-Box** and performs **Inverse ShiftRows** and **Inverse MixColumns**.

AES Example

Suppose we want to encrypt the plaintext “HELLO” with a 128-bit AES key.

1. The plaintext is first converted into a **16-byte block** (padded if necessary).
2. The AES algorithm transforms the block through multiple rounds of substitution, shifting, mixing, and key addition.
3. The final output is a **ciphertext block of 128 bits**.

Code:

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import base64

def pad(text):
    return text + (16 - len(text) % 16) * chr(16 - len(text) % 16)

def unpad(text):
```

```
return text[:-ord(text[-1])]

def encrypt_AES(plaintext, key):
    cipher = AES.new(key, AES.MODE_CBC)
    ct_bytes = cipher.encrypt(pad(plaintext).encode('utf-8'))
    iv = base64.b64encode(cipher.iv).decode('utf-8')
    ct = base64.b64encode(ct_bytes).decode('utf-8')
    return iv, ct

def decrypt_AES(iv, ciphertext, key):
    iv = base64.b64decode(iv)
    ct = base64.b64decode(ciphertext)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    pt = unpad(cipher.decrypt(ct).decode('utf-8'))
    return pt

if __name__ == "__main__":
    while True:
        print("\n===== AES Algorithm Menu =====")
        print("1. Encrypt")
        print("2. Decrypt")
        print("3. Exit")
        choice = input("Enter choice: ")

        if choice == "1":
            key = get_random_bytes(16)
            text = input("Enter plaintext: ")
            iv, cipher = encrypt_AES(text, key)
            print("Key:", base64.b64encode(key).decode('utf-8'))
            print("IV :", iv)
            print("Ciphertext:", cipher)

        elif choice == "2":
            key = input("Enter Base64 Key: ")
            iv = input("Enter IV: ")
            ciphertext = input("Enter Ciphertext: ")

            key = base64.b64decode(key)
            decrypted = decrypt_AES(iv, ciphertext, key)
            print("Decrypted Text:", decrypted)

        elif choice == "3":
            print("Exiting AES Program.")
            break
        else:
            print("Invalid choice. Try again.")
```

OUTPUT :

```
02\Assignment02.py"
●
===== AES Algorithm Menu =====
1. Encrypt
2. Decrypt
3. Exit
Enter choice: 1
Enter plaintext: Hello AES
Key: UxMwufL54g0I0j0CTBy2GA==
IV : 1G6jS8HSD7TvWsIPyATY0w==
Ciphertext: 1nCWTJ1939vDhHyr2gs7sg==

===== AES Algorithm Menu =====
1. Encrypt
2. Decrypt
3. Exit
Enter choice: 2
Enter Base64 Key: UxMwufL54g0I0j0CTBy2GA==
Enter IV: 1G6jS8HSD7TvWsIPyATY0w==
Enter Ciphertext: 1nCWTJ1939vDhHyr2gs7sg==
Decrypted Text: Hello AES

===== AES Algorithm Menu =====
1. Encrypt
2. Decrypt
3. Exit
Enter choice: 3
Exiting AES Program.
○ PS C:\Users\CHANDRAKANT THAKARE\Desktop\TY IS Assignments>
```