**Cryptography and Network Security**

**Lab**

**Assignment No. 6**

**Name : Sourabh Shankar Patil**

**PRN : 21510045**

**Batch : B2**

**Advanced Encryption Standard (AES)**

The **Advanced Encryption Standard (AES)** is a widely used symmetric-key block cipher that was established by the National Institute of Standards and Technology (NIST) in 2001. It is the successor to the Data Encryption Standard (DES) and is known for its high level of security and efficiency.

**Key Features:**

* **Block Size**: AES operates on fixed block sizes of 128 bits.
* **Key Lengths**: AES supports three key lengths: 128, 192, and 256 bits, allowing for different levels of security.
* **Symmetric Encryption**: The same key is used for both encryption and decryption, making key management critical.

**Structure:**

* **Substitution-Permutation Network (SPN)**: AES uses a series of well-defined transformations that include:
  + **SubBytes**: A non-linear substitution step where each byte is replaced with another byte using a fixed substitution table (S-box).
  + **ShiftRows**: A transposition step where each row of the state is shifted cyclically to the left.
  + **MixColumns**: A mixing operation that combines the bytes of each column.
  + **AddRoundKey**: A key addition step where a round key is combined with the state using bitwise XOR.
* **Rounds**: AES uses a variable number of rounds depending on the key length:
  + 10 rounds for 128-bit keys
  + 12 rounds for 192-bit keys
  + 14 rounds for 256-bit keys

**Security:**

* AES is considered highly secure and is used in various applications, including data encryption, secure communications, and VPNs.
* It has been extensively analyzed and is resistant to most known cryptographic attacks, making it a standard choice for secure data transmission.

**Applications:**

* AES is widely implemented in software and hardware across various industries, including banking, telecommunications, and government.
* It is also used in protocols such as SSL/TLS for secure internet communication.

#include <iostream>

#include <vector>

#include <iomanip>

using namespace std;

// AES S-box table

unsigned char sbox[256] = {

    0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5,

    0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab, 0x76,

    0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0,

    0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0,

    0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc,

    0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31, 0x15,

    0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a,

    0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75,

    0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0,

    0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f, 0x84,

    0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b,

    0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf,

    0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85,

    0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f, 0xa8,

    0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5,

    0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2,

    0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17,

    0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19, 0x73,

    0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88,

    0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb,

    0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c,

    0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79,

    0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9,

    0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08,

    0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6,

    0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a,

    0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e,

    0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e,

    0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94,

    0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf,

    0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68,

    0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16};

void AES\_encrypt(vector<unsigned char> &state)

{

    for (int i = 0; i < 16; i++)

    {

        state[i] = sbox[state[i]]; // SubBytes operation

    }

}

int main()

{

    vector<unsigned char> plaintext = {0x32, 0x43, 0xf6, 0xa8, 0x88, 0x5a, 0x30, 0x8d, 0x31, 0x31, 0x98, 0xa2, 0xe0, 0x37, 0x07, 0x34};

    cout << "Plaintext: ";

    for (unsigned char c : plaintext)

    {

        cout << hex << setw(2) << setfill('0') << (int)c << " ";

    }

    cout << endl;

    AES\_encrypt(plaintext);

    cout << "Encrypted text: ";

    for (unsigned char c : plaintext)

    {

        cout << hex << setw(2) << setfill('0') << (int)c << " ";

    }

    cout << endl;

    return 0;

}

Output:

