SEM - VII - 2022-23

CNS Lab

B3 - 2019BTECS00094 - Sweety Shrawan Gupta

Assignment 7

AES - Advanced Encryption Standard

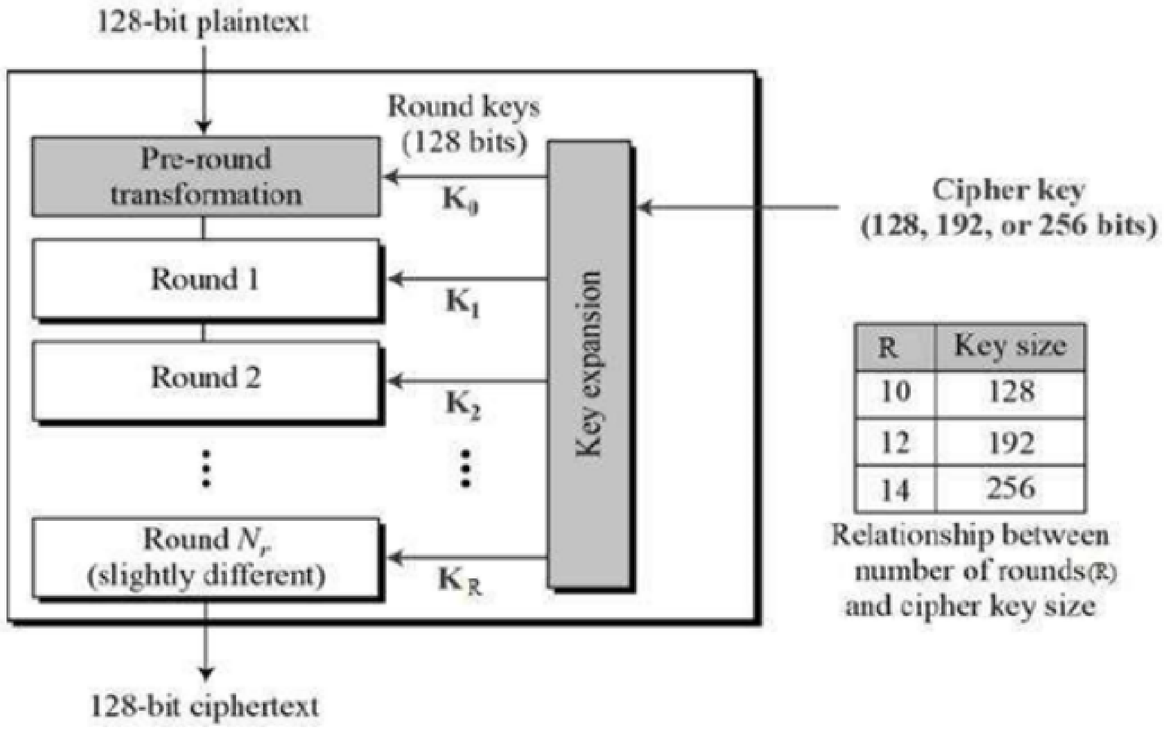
Objective: To study and implement encryption and decryption using AES

Theory:

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

AES is a block cipher. The key size can be 128/192/256 bits. It encrypts data in blocks of 128 bits each. That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

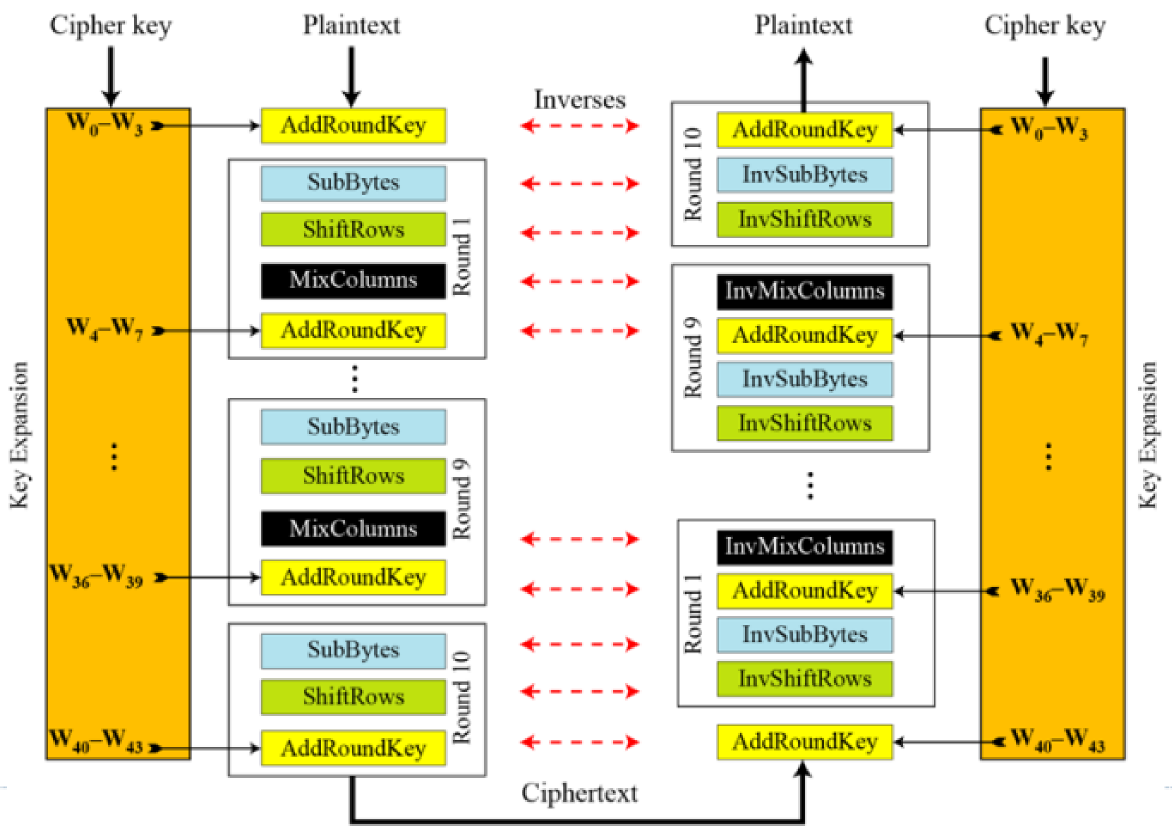
The schematic of AES structure



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Overall Structure

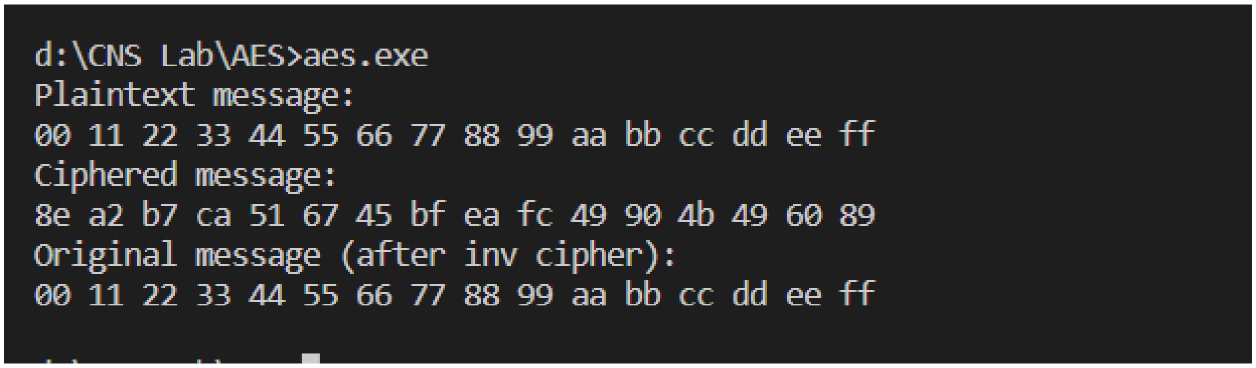


Code:

```
/*
 * Advanced Encryption Standard
 * @author Dani Huertas
 * @email huertas.dani@gmail.com
 *
 * Based on the document FIPS PUB 197
 */
#include <stdio.h>
```

```
#include "aes.h"

int main() {

    uint8_t i;

    /*
     * Appendix A - Key Expansion Examples
     */

    /* 128 bits */
    /* uint8_t key[] = {
        0x2b, 0x7e, 0x15, 0x16,
        0x28, 0xae, 0xd2, 0xa6,
        0xab, 0xf7, 0x15, 0x88,
        0x09, 0xcf, 0x4f, 0x3c}; */

    /* 192 bits */
    /* uint8_t key[] = {
        0x8e, 0x73, 0xb0, 0xf7,
        0xda, 0x0e, 0x64, 0x52,
        0xc8, 0x10, 0xf3, 0x2b,
        0x80, 0x90, 0x79, 0xe5,
        0x62, 0xf8, 0xea, 0xd2,
        0x52, 0x2c, 0x6b, 0x7b}; */

    /* 256 bits */
    /* uint8_t key[] = {
        0x60, 0x3d, 0xeb, 0x10,
        0x15, 0xca, 0x71, 0xbe,
        0x2b, 0x73, 0xae, 0xf0,
        0x85, 0x7d, 0x77, 0x81,
        0x1f, 0x35, 0x2c, 0x07,
        0x3b, 0x61, 0x08, 0xd7,
        0x2d, 0x98, 0x10, 0xa3,
        0x09, 0x14, 0xdf, 0xf4};
    */

    /* uint8_t in[] = {
        0x32, 0x43, 0xf6, 0xa8,
    0x88, 0x5a, 0x30, 0x8d,
    0x31, 0x31, 0x98, 0xa2,
    0xe0, 0x37, 0x07, 0x34}; // 128
*/

/*
 * Appendix C - Example Vectors
 */

/* 128 bit key */
/* uint8_t key[] = {
    0x00, 0x01, 0x02, 0x03,
    0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b,
    0x0c, 0x0d, 0x0e, 0x0f}; */

/* 192 bit key */
/* uint8_t key[] = {
    0x00, 0x01, 0x02, 0x03,
    0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b,
    0x0c, 0x0d, 0x0e, 0x0f,
    0x10, 0x11, 0x12, 0x13,
    0x14, 0x15, 0x16, 0x17}; */

/* 256 bit key */
uint8_t key[] = {
    0x00, 0x01, 0x02, 0x03,
    0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b,
    0x0c, 0x0d, 0x0e, 0x0f,
    0x10, 0x11, 0x12, 0x13,
    0x14, 0x15, 0x16, 0x17,
    0x18, 0x19, 0x1a, 0x1b,
    0x1c, 0x1d, 0x1e, 0x1f};

uint8_t in[] = {
    0x00, 0x11, 0x22, 0x33,
    0x44, 0x55, 0x66, 0x77,
    0x88, 0x99, 0xaa, 0xbb,
        0xcc, 0xdd, 0xee, 0xff};

    uint8_t out[16]; // 128

    uint8_t *w; // expanded key

    w = aes_init(sizeof(key));

    aes_key_expansion(key, w);

    printf("Plaintext message:\n");
    for (i = 0; i < 4; i++) {
        printf("%02x %02x %02x %02x ", in[4*i+0], in[4*i+1], in[4*i+2],
in[4*i+3]);
    }

    printf("\n");

    aes_cipher(in /* in */, out /* out */, w /* expanded key */);

    printf("Ciphered message:\n");
    for (i = 0; i < 4; i++) {
        printf("%02x %02x %02x %02x ", out[4*i+0], out[4*i+1], out[4*i+2],
out[4*i+3]);
    }

    printf("\n");

    aes_inv_cipher(out, in, w);

    printf("Original message (after inv cipher):\n");
    for (i = 0; i < 4; i++) {
        printf("%02x %02x %02x %02x ", in[4*i+0], in[4*i+1], in[4*i+2],
in[4*i+3]);
    }

    printf("\n");

    free(w);
    return 0;
}
```

Output:



```
d:\CNS Lab\AES>aes.exe
Plaintext message:
00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
Ciphered message:
8e a2 b7 ca 51 67 45 bf ea fc 49 90 4b 49 60 89
Original message (after inv cipher):
00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff
```