COSC 3P93 PROJECT STEP 2

AES Encryption

Delivered by -

- Fahad Ansar
- Paramvir Singh

INTRODUCTION

AES algorithm is an acronym for Advanced Encryption Standard algorithm. Its original name is Rijndael. AES is a subset of the Rijndael block which was developed by two Belgian cryptographers Vincent and Joan. It is based on a design principle known as substitution permutation network which works on both software and hardware.

AES has a fixed size of 128 bits with a key sizes of 128 bits, 192 bits and 256 bits. It operates on a 4x4 matrix which is treated as an array of bytes.
AES has a fixed number of transformation rounds where steps are repeated using the last state of the matrix as an input. There are fixed number of rounds depending on size of the key,

- For 128 bit key its 10 rounds
- For 192 bit key its 12 rounds
- For 256 bit key its 14 rounds

ENCRYPTION PROCESS

AES-128 uses a 128 bit key for encryption.

Key and plain text message are stored as hexadecimal values ranging from 0x00 to 0xff.

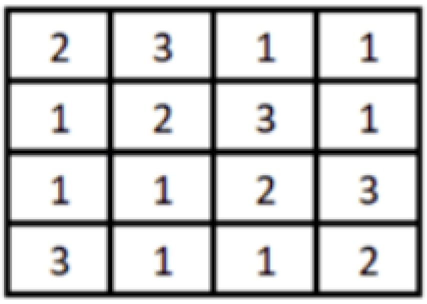
Both key and plain text message are stored as a 4*4 matrix. Plain text message matrix is referred to as a state-matrix while it is in the process of encryption.

We perform a xor operation on the initial key (provided by the user) and the plain text message, before the 1st round. The result of this operation known as the state matrix is passed on.

Each round has 4 operations (is carried out 10 times for AES-128)

- Sub bytes
    - Substitutes the values of the state-matrix from the s-box table.
    - The first 4 bytes of hexadecimal numbers are used to select the row number from the sub-byte table.
    - Another 4 bytes of the hexadecimal numbers used to select the column number from the sub-byte table.
    - For example, 0xa6 means row number a and column number 6.
    - The resultant of this function is passed on to the next function.
- Shift rows
    - All the rows of the state-matrix are shifted towards left
    - The first row is shifted zero times. The second row is shifted twice and so on.
    - The resultant is passed onto the next function.
- Mix columns (is not performed for the 10th round)

    - Each column is multiplied with Rijndael-Mix-Column-Table.



2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

Rijndael Mix Column Table

        - The multiplication step consists of addition [XOR] and multiplication [modulo of the polynomial]. Each byte is considered as a polynomial of order x^7.
        - There is a chance of an overflow which occurs in the case when the result of multiplication is larger than OxFF. To tackle this overflow, we perform XOR with Ox1B on that byte

    - Resultant of this step is passed onto the next function.
- Add round key
    - Xor operation is carried out between the state-matrix and the first key from the key-expansion list.

The encrypted text is obtained.

PSEUDO CODE

```
/**
* Used to encrypt a block of text of 128 bits. For message bigger
* than 128 bits we can send multiple blocks of plain text of 128
* bits.
*/

key[4][4] = char_to_hex(user_key)
msg[4][4] = char_to_hex(user_message)

round_keys = key_expansion(key)

state = add_round_key(key, msg)

for i 0 to 9
      state = sub_bytes(state)
```

```
     state = shift_rows(state)
     state = mix_columns(state)
     state = add_round_key(state, round_keys[i])

state = sub_bytes(state)
state = shift_rows(state)
cipher = add_round_key(state)
```

RUNNING THE PROGRAM

Application uses C++ 14 Standard and CMake 3.17

Program was written in CLion IDE. If you are using the same IDE hitting the ***run*** button will execute the code. If you wish to run the code from the console, just go to the directory ***cmake-build-debug***, then enter ***make*** to build the project and enter ***./AES*** to run the application.

To use a key of your choice, enter the desired key in **src/key.txt**. Key should be 16 characters long.

To use a test message of your choice, enter the desired message in **src/message.txt**. Text message can be as long as possible.

Encrypted message is saved in **src/encrypted.txt**.