Name : Harsh Verdhan Singh
Roll No : 20mcs009

ReadMe :

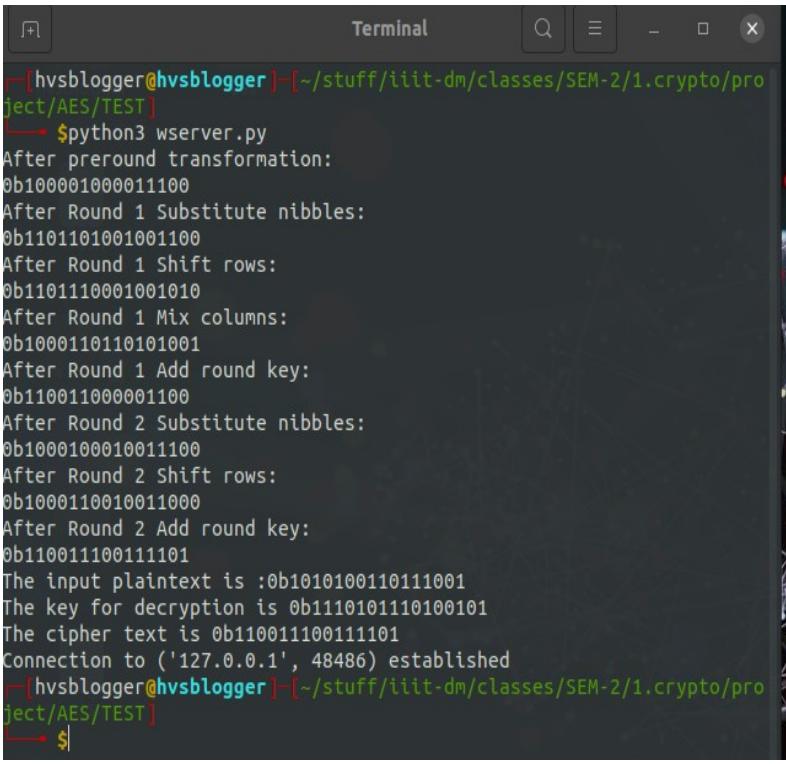
In this zip file there are two python file client.py and server.py. Client.py is used for client side socket programming and server.py is used for server side socket programming.

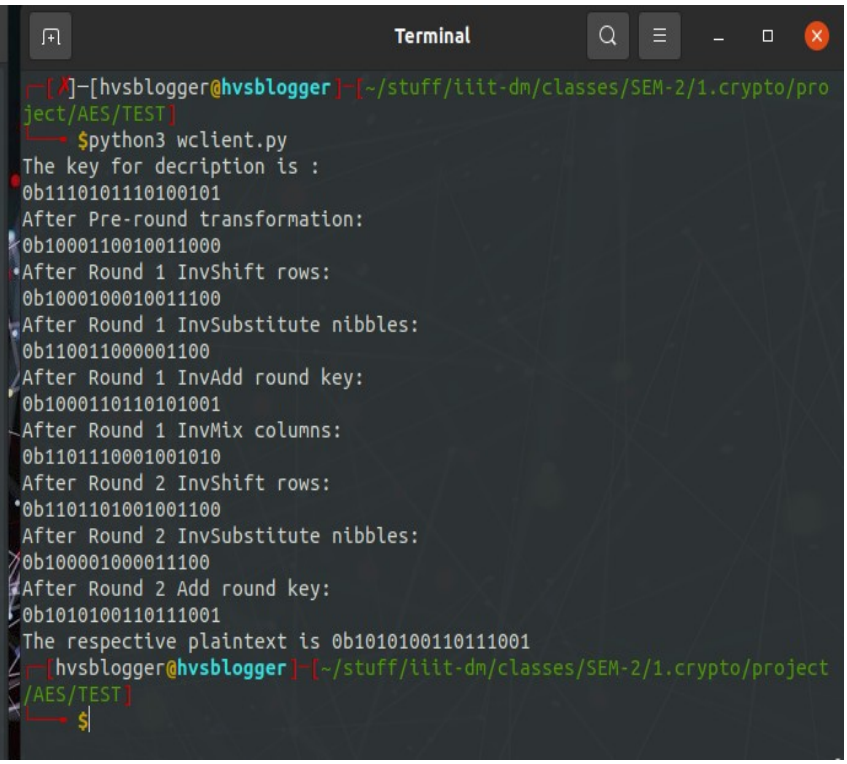
In both file the implementation of AES is done as given in the assignment.

For execution initially the server file must be run else on running the client file first it will show error of refused request.

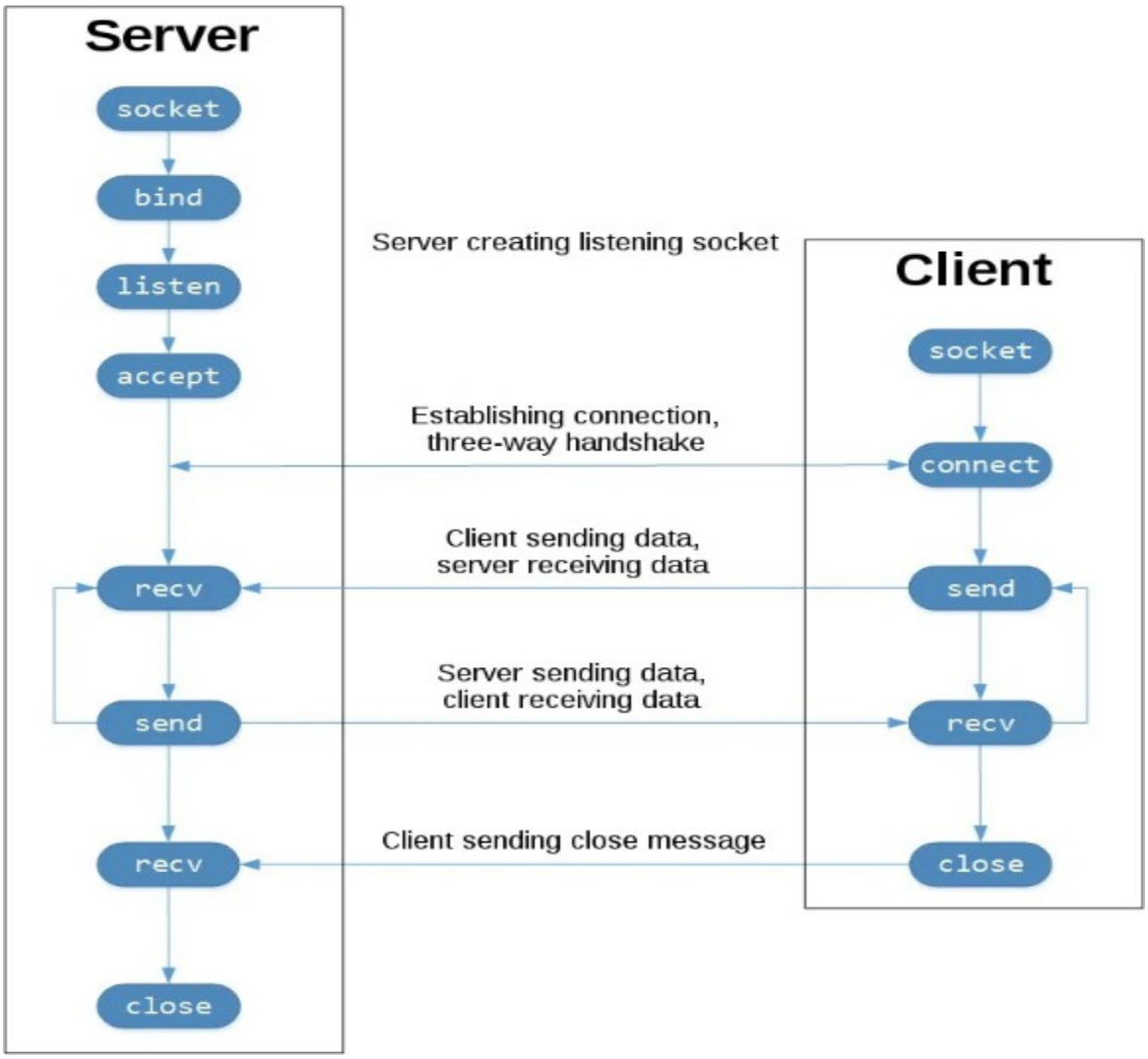
Here the key and plaintext both are of 16 bit bin format.

The execution step is shown blew : Here I also print each step that what each step is doing .

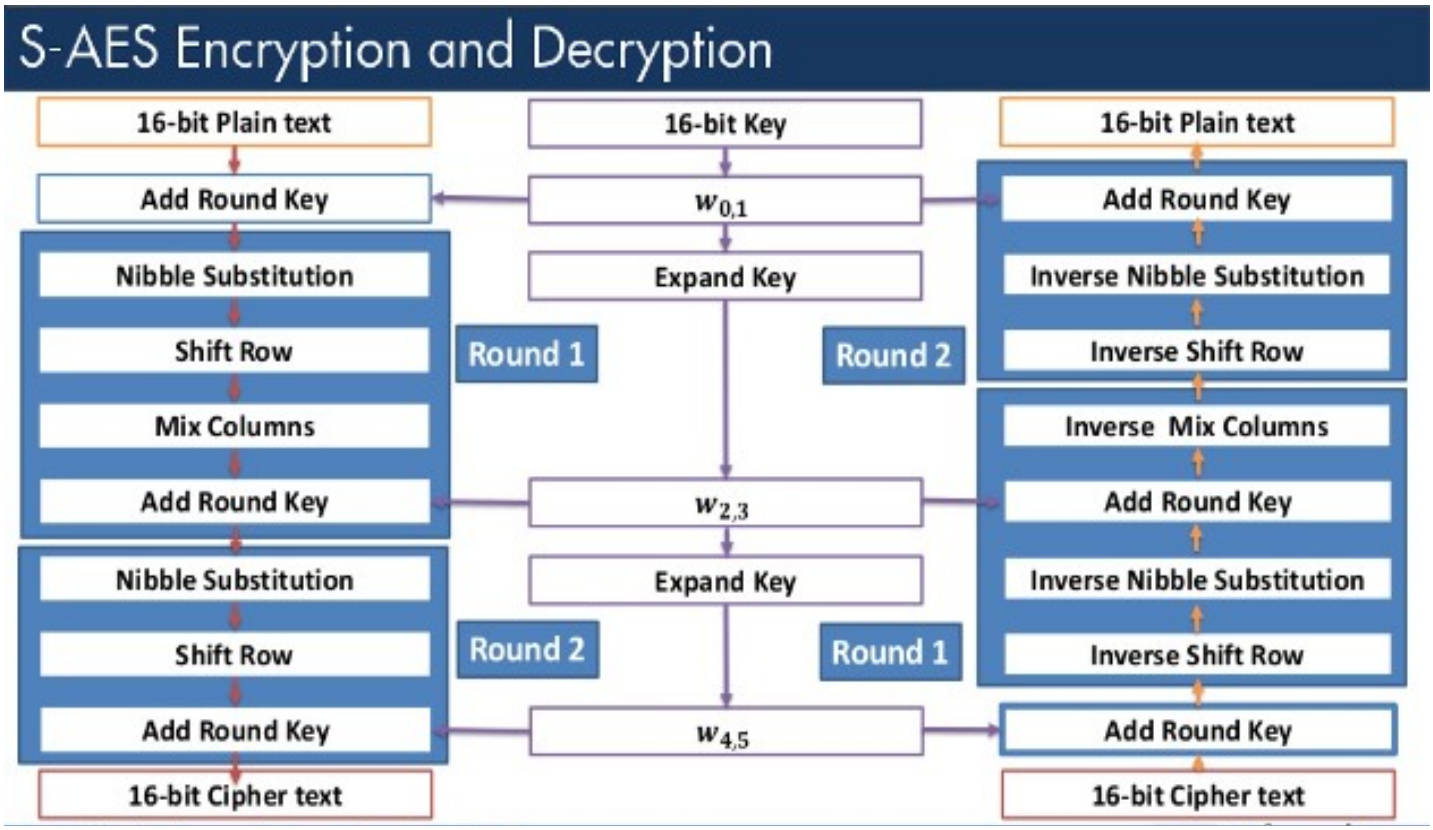




Socket programming



AES Implementation



16-bit block

16-bit key

4 x 4 S-box

Field 16f6

Modulus X^4+X+1

2 rounds

sbox ; substitution box and substitution Inverse are initialized.

sub_word() is defined to Substitute word Block with the help of sbox, subtitute nibbles to another nibbles. Each nibble is replaced by nibble indexed by row (left 4--bits) & column (right 4--bits) of a 4x4 table

gf_mult(a, b): Galois field multiplication of a and b in GF(2^4) / x^4 + x + 1

a&b are respective numbers and will return the product of both a&b under GF(2^4)

int_to_state(n): is converting an integer of 2B into a 4-element vector (state matrix) and it returns the state corresponding to the integer value

state_to_int(): this is reverse of int_to_state(). It will convert the 4 element vector into a 2B int and return an integer corresponding to that state.

add_round_key(s1, s2): Add round keys in GF(2^4) ,(s1 -> First number),(s2 -> Second number) and it returns the Addition of both under GF(2^4)

sub_nibbles(sbox, state): block for Nibble substitution (sbox for Substitution box to use for transformation) (State to perform sub nibbles transformation on , return the Resultant state

shift_rows(state): Shift rows and inverse shift rows of state matrix (same) , State to perform shift rows transformation and returns Resultant state)

Mix Columns(state): Apply the matrix multiplication with the constant matrix, Me, using GF(24). For GF(24), the addition operation is simply an XOR, and for the multiplication operation you can use a lookup table.

inverse_mix_columns(state): Inverse mix columns transformation on state matrix, State to perform inverse mix columns transformation and Resultant the state

encrypt(plaintext,key): this function takes 2 arguments plaintext: the normal 16bit message and key 16bit. Then it calls the function one by one and ecrypts the message.

Decrypt(ciphertext,key): this will takes 2 argument ciphertext ,key and returns the actual plaintext.