Practical 10

Problem statement : Implement encryption and decryption using AES scheme.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <mcrypt.h>

#include <math.h>

#include <stdint.h>

int encrypt(

    void\* buffer,

    int buffer\_len,

    char\* IV,

    char\* key,

    int key\_len

){

  MCRYPT td = mcrypt\_module\_open("rijndael-128", NULL, "cbc", NULL);

  if (td == MCRYPT\_FAILED) {

    return 1;

  }

  int blocksize = mcrypt\_enc\_get\_block\_size(td);

  if (buffer\_len % blocksize != 0) {

    mcrypt\_module\_close(td);

    return 1;

  }

  mcrypt\_generic\_init(td, key, key\_len, IV);

  mcrypt\_generic(td, buffer, buffer\_len);

  mcrypt\_generic\_deinit(td);

  mcrypt\_module\_close(td);

  return 0;

}

int decrypt(

    void\* buffer,

    int buffer\_len,

    char\* IV,

    char\* key,

    int key\_len

){

  MCRYPT td = mcrypt\_module\_open("rijndael-128", NULL, "cbc", NULL);

  if (td == MCRYPT\_FAILED) {

    return 1;

  }

  int blocksize = mcrypt\_enc\_get\_block\_size(td);

  if (buffer\_len % blocksize != 0) {

    mcrypt\_module\_close(td);

    return 1;

  }

  mcrypt\_generic\_init(td, key, key\_len, IV);

  mdecrypt\_generic(td, buffer, buffer\_len);

  mcrypt\_generic\_deinit(td);

  mcrypt\_module\_close(td);

  return 0;

}

void display(char\* ciphertext, int len){

  for (int v = 0; v < len; v++) {

    printf("%d ", (unsigned char)ciphertext[v]);

  }

  printf("\n");

}

int main()

{

  char\* plaintext = "test text 123";

  char\* IV = "AAAAAAAAAAAAAAAA";

  char\* key = "0123456789abcdef";

  int keysize = 16; /\* 128 bits \*/

  int buffer\_len = 16;

  char\* buffer = calloc(1, buffer\_len);

  if (buffer == NULL) {

    perror("calloc");

    return 1;

  }

  strncpy(buffer, plaintext, buffer\_len - 1);

  buffer[buffer\_len - 1] = '\0'; // Ensure null-termination

  printf("==C==\n");

  printf("plain:   %s\n", plaintext);

  if (encrypt(buffer, buffer\_len, IV, key, keysize) != 0) {

    fprintf(stderr, "Encryption failed\n");

    free(buffer);

    return 1;

  }

  printf("cipher:  "); display(buffer, buffer\_len);

  if (decrypt(buffer, buffer\_len, IV, key, keysize) != 0) {

    fprintf(stderr, "Decryption failed\n");

    free(buffer);

    return 1;

  }

  printf("decrypt: %s\n", buffer);

  free(buffer);

  return 0;

}