Program 40 C program for ECB, CBC, and CFB modes, the plaintext must be a sequence of one or more

complete data blocks

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define BLOCK\_SIZE 8

void encryptECB(char \*plaintext, char \*key, int len);

void encryptCBC(char \*plaintext, char \*key, char \*iv, int len);

void encryptCFB(char \*plaintext, char \*key, char \*iv, int len);

int main() {

char plaintext[] = "ABCDEFGH12345678";

char key[] = "mysecret";

char iv[] = "initvect";

printf("Original plaintext: %s\n\n", plaintext);

// ECB Mode

encryptECB(plaintext, key, strlen(plaintext));

printf("ECB Encrypted: %s\n\n", plaintext);

// Reset plaintext

strcpy(plaintext, "ABCDEFGH12345678");

// CBC Mode

encryptCBC(plaintext, key, iv, strlen(plaintext));

printf("CBC Encrypted: %s\n\n", plaintext);

// Reset plaintext

strcpy(plaintext, "ABCDEFGH12345678");

// CFB Mode

encryptCFB(plaintext, key, iv, strlen(plaintext));

printf("CFB Encrypted: %s\n", plaintext);

return 0;

}

void encryptECB(char \*plaintext, char \*key, int len) {

for (int i = 0; i < len; i += BLOCK\_SIZE) {

for (int j = 0; j < BLOCK\_SIZE; j++) {

plaintext[i + j] ^= key[j];

}

}

}

void encryptCBC(char \*plaintext, char \*key, char \*iv, int len) {

for (int i = 0; i < len; i += BLOCK\_SIZE) {

for (int j = 0; j < BLOCK\_SIZE; j++) {

plaintext[i + j] ^= iv[j];

}

encryptECB(plaintext + i, key, BLOCK\_SIZE);

memcpy(iv, plaintext + i, BLOCK\_SIZE);

}

}

void encryptCFB(char \*plaintext, char \*key, char \*iv, int len) {

for (int i = 0; i < len; i += BLOCK\_SIZE) {

encryptECB(iv, key, BLOCK\_SIZE);

for (int j = 0; j < BLOCK\_SIZE; j++) {

plaintext[i + j] ^= iv[j];

}

memcpy(iv, plaintext + i, BLOCK\_SIZE);

}

}

output;

Original plaintext: ABCDEFGH12345678

ECB Encrypted: 9x3=C)?E2>@5

CBC Encrypted: BmpWptg@HR2k(Fc

CFB Encrypted: iuXQi9HLt:WxW.8