20. Write a C program for ECB mode, if there is an error in a block of the transmitted ciphertext, only the corresponding plaintext block is affected. However, in the CBC mode, this error propagates. For example, an error in the transmitted C1 obviously corrupts P1 and P2.

a. Are any blocks beyond P2 affected?

b. Suppose that there is a bit error in the source version of P1. Through how many ciphertext blocks is this error propagated? What is the effect at the receiver?

#include <stdio.h>

#include <string.h>

#define BLOCK\_SIZE 8

void xor\_encrypt\_block(unsigned char \*block, unsigned char \*key) {

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

block[i] ^= key[i];

}

}

int main() {

unsigned char key[BLOCK\_SIZE] = {1, 2, 3, 4, 5, 6, 7, 8};

unsigned char plaintext[] = "ThisIsBlock1ThisIsBlock2";

unsigned char ciphertext[32];

unsigned char decrypted[32];

int len = strlen((char \*)plaintext);

int blocks = len / BLOCK\_SIZE;

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

for (int i = 0; i < blocks; i++) {

memcpy(&ciphertext[i \* BLOCK\_SIZE], &plaintext[i \* BLOCK\_SIZE], BLOCK\_SIZE);

xor\_encrypt\_block(&ciphertext[i \* BLOCK\_SIZE], key);

}

printf("Ciphertext (hex): ");

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

printf("%02x", ciphertext[i]);

}

printf("\n");

for (int i = 0; i < blocks; i++) {

memcpy(&decrypted[i \* BLOCK\_SIZE], &ciphertext[i \* BLOCK\_SIZE], BLOCK\_SIZE);

xor\_encrypt\_block(&decrypted[i \* BLOCK\_SIZE], key);

}

decrypted[blocks \* BLOCK\_SIZE] = '\0';

printf("Decrypted Plaintext: %s\n", decrypted);

return 0;

}

OUTPUT

