Министерство образования и науки Украины

Харьковский национальный университет радиоэлектроники

Кафедра БИТ

Отчет

По лабораторной работе по ОАСК/МПАК

Тема «Режим OFB»

Выполнили: Проверил:

ст.гр БИКСм-18-1 Мельникова О. А.

До Тхао

Харьков 2018

#include "Encrypt.hpp"

void BlockEncrypt128128(uchar \*state, uchar \*\*rkey)

{

const int len8 = 16;

AddrKey(state, len8, state, rkey[0]);

for (int i = 1; i < 10; i++) {

SubBytes(state, len8, s\_box, state);

ShiftRow(state, len8);

MixColum(state, len8);

XorKey(state, len8, rkey[i]);

}

SubBytes(state, len8, s\_box, state);

ShiftRow(state, len8);

MixColum(state, len8);

AddrKey(state, len8, state, rkey[10]);

}

void OFB128128(uchar \*src, unsigned int lenSrc, uchar \*&dst)

{

const int LEN8 = 16;

dst = new uchar[lenSrc];

uchar IV[LEN8] = { 0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x1A,0x1B,0x1C,0x1D,0x1E,0x1F },

Key[LEN8] = { 0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0A,0x0B,0x0C,0x0D,0x0E,0x0F };

uchar \*\*rkey = new uchar\*[11];

for (int i = 0; i < 11; i++) rkey[i] = new uchar[16];

GenRoundKeys128128(rkey, Key);

int i, j, k,

remBlock = lenSrc % LEN8,

divBlock = lenSrc - remBlock;

for (i = 0; i < divBlock; i += LEN8) {

BlockEncrypt128128(IV, rkey);

for (j = LEN8 - 1, k = i; j >= 0; j--, k++) {

dst[k] = IV[j] ^ src[k];

}

}

BlockEncrypt128128(IV, rkey);

for (j = remBlock, k = i; j >= 0; j--, k++) {

dst[k] = IV[j] ^ src[k];

}

}

#include <iostream>

#include <cstring>

#include <iomanip>

#include <fstream>

using namespace std;

#include "Encrypt.hpp"

int readFile(uchar \*&plaintext, const char\* filename)

{

FILE \*fin;

fopen\_s(&fin, filename, "rb");

if (!fin) return 0;

fseek(fin, 0, SEEK\_END);

int len = ftell(fin);

fseek(fin, 0, SEEK\_SET);

plaintext = new uchar[len];

char \*str = new char[len];

fread(plaintext, 1, len, fin);

fclose(fin);

return len;

}

int write2File(uchar \*text, int len, const char\* filename)

{

FILE \*fout;

fopen\_s(&fout, filename, "wb");

if (!fout) return 0;

fwrite(text, 1, len, fout);

fclose(fout);

return len;

}

void testOFB()

{

uchar \*M = new uchar[1];

uchar \*C = new uchar[1];

uchar \*M1 = new uchar[1];

uchar \*C1 = new uchar[1];

cout << "Encrypt: " << endl;

int lenM = readFile(M, "Plaintext.txt");

cout << "lenM: " << lenM << endl;

cout << "M : ";

cout << M << endl << endl;

OFB128128(M, lenM, C);

cout << "==>C : "; printuc(C, lenM);

write2File(C, lenM, "Ciphertext.txt");

cout << endl << endl;

cout << "Decrypt: " << endl;

int lenC = readFile(C1, "Ciphertext.txt");

cout << "lenC: " << dec << lenC << endl;

cout << "C1: "; printuc(C1, lenC); cout << endl;

OFB128128(C1, lenC, M1);

cout << "==>M1: ";

cout << M1 << endl << endl;

}

int main()

{

testOFB();

system("pause");

return 0;

}

void printuc(uchar \*Src, int len) {

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

cout << hex << setw(2) << setfill('0') << uppercase << (int)Src[i];

}

cout << endl;

}



