УO «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Лабораторная работа

**«Криптография в С++»**

Выполнил студент

3 курса, 4 группы

Чаевский А. В.

Проверила

Герман Ю. О.

Минск 2019

# **1 Постановка задачи**

Цель: Изучить базовые принципы криптографии. ТАЙМЕРЫ И КЛАВА/МЫШЬ ОТЧЁТ СДЕЛАТЬ

# **2 Краткие сведения**

Для шифрования нужно получить криптопровайдер. Криптопровайдер – это объект, который обеспечивает методы шифрования и дешифрования. Мы получаем его таким образом:

if (!CryptAcquireContext(&hProv, NULL, NULL,

PROV\_RSA\_FULL, CRYPT\_VERIFYCONTEXT))

Здесь первый аргумент получает ссылку на криптопровайдер. Указывается, что он обеспечивает шифрование по методу RSA. Детали этого метода шифрования мы оставляем без объяснений. Для шифрования нужно получить ключ. Ключ генерирует система таким образом:

CryptGenKey(hProv, CALG\_RC4,

CRYPT\_EXPORTABLE, &hSessionKey);

Ключ определяется в адресной переменной hSessionKey.

Шифрование выполняется по команде

if (!CryptEncrypt(hSessionKey, 0, true, 0, (BYTE\*)string,

&count, strlen(string)))

Шифруется строка string с помощью подготовленного ключа. Количество шифруемых байтов задается переменной count.

Программа расшифровки вызывается таким образом

if(CryptDecrypt(

hSessionKey,

0,

TRUE,

0,

pbBuffer,

&count))

Расшифрованная строка помещается в буфер pbBuffer.

Полный текст программы имеет следующий вид:

// CRYPTO.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <stdio.h>

#include <windows.h>

#include <wincrypt.h>

#include <cryptuiapi.h>

#include <iostream>

#include <tchar.h>

#define KEYLENGTH 0x00800000

#define ENCRYPT\_BLOCK\_SIZE 8

#pragma comment (lib, "crypt32.lib")

#pragma comment (lib, "cryptui.lib")

int \_tmain(int argc, \_TCHAR\* argv[])

{

HCRYPTPROV hProv;

HCRYPTKEY hSessionKey;

// Получение контекста криптопровайдера

if (!CryptAcquireContext(&hProv, NULL, NULL,

PROV\_RSA\_FULL, CRYPT\_VERIFYCONTEXT))

{

std::cout << "Error in Crypto1" << std::endl;

return 1;

}

std::cout << "Cryptographic provider initialized" << std::endl;

// Генерация сессионного ключа

CryptGenKey(hProv, CALG\_RC4,

CRYPT\_EXPORTABLE, &hSessionKey);

//if (!CryptGenKey(hProv, CALG\_RC4,

// CRYPT\_ENCRYPT | CRYPT\_DECRYPT, &hSessionKey))

//{

// Error("CryptGenKey");

// return;

//}

std::cout << "Session key generated" << std::endl;

// Данные для шифрования

char string[]="Test";

DWORD count=strlen(string);

DWORD delen = strlen(string);

PBYTE pbBuffer=NULL;

DWORD dwBufferLen = count + ENCRYPT\_BLOCK\_SIZE - (count % ENCRYPT\_BLOCK\_SIZE);

// Allocate memory

pbBuffer = (BYTE \*)malloc(dwBufferLen);

// Шифрование данных

if (!CryptEncrypt(hSessionKey, 0, true, 0, (BYTE\*)string,

&count, strlen(string)))

{

std::cout << "Error in Crypto2" << std::endl;

return 1;

}

std::cout << "Encryption completed" << std::endl;

// Тестовый вывод на экран

std::cout << "Encrypted string: " << string << std::endl;

////+++++++++++++

memcpy(pbBuffer, string, delen);

if(CryptDecrypt(

hSessionKey,

0,

TRUE,

0,

pbBuffer,

&count))

{

printf("After decryption: %s\n", pbBuffer);

//std::cout << "\nDecrypted text = ";

//std::cout<<pbBuffer<<std::endl;

}

//------

getchar();

return 0;

}

Скриншот такой:

# 

В настоящей лабораторной работе нужно ввести пароль, который забивается звездочками. Костяк работы представлен следующей не совсем верной программой. Надо дать студентам задачу эту программу исправить. Во-вторых, эту программу нужно использовать как развитие предыдущей лабы, где пароль вводился нормальным текстом (но не забивался звездочками) и сравнивался с зашифрованным паролем из файла. Таким образом, студент должен получить законченный и достаточно полезный модуль для ввода пароля с консоли.

Кроме того, в данной работе пароль фиксирован – stern. Его нужног сделатьлюбым, т.е. читать любую строку из пяти символов.

// pasword\_curs.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <windows.h>

#include <cstdio>

#include <cstdlib>

#include "string.h"

#include "conio.h"

#define STRLEN(x) (sizeof(x)/sizeof(TCHAR)-1)

using namespace std;

HANDLE output;

HANDLE input;

COORD cursor;

int x=10;

int y=3;

const TCHAR szMsg[]=L"Input password:";

TCHAR szName[1];

char carr[4];

DWORD dwCount=0;

int \_tmain(int argc, \_TCHAR\* argv[])

{

output = GetStdHandle(STD\_OUTPUT\_HANDLE);

input= GetStdHandle(STD\_INPUT\_HANDLE);

cursor.X = x;

cursor.Y = y;

SetConsoleCursorPosition(output,cursor);

WriteConsole(output,&szMsg,STRLEN(szMsg),&dwCount,NULL);

cursor.Y++;

SetConsoleCursorPosition(output,cursor);

char ch;

int i=0;

bool priz=true;

while(i<4)

{

//ReadConsole(input,&szName,STRLEN(szName),&dwCount,NULL);

ch=getch();

carr[i]=ch;

switch (i)

{

case 0:

if(!(ch=='s'))

{

priz=false;

break;

}

break;

case 1:

if(!(ch=='e'))

{

priz=false;

break;

}

break;

case 2:

if(!(ch=='r'))

{

priz=false;

break;

}

break;

case 3:

if(!(ch=='n'))

{

priz=false;

break;

}

break;

default:

{

priz=false;

break;

}

}

ch='\*';

cursor.X++;

SetConsoleCursorPosition(output,cursor);

printf("%c",ch);

i++;

}

if (priz)

cout << "OK!"<<std::endl;

else

cout << "GAME OVER!"<<std::endl;

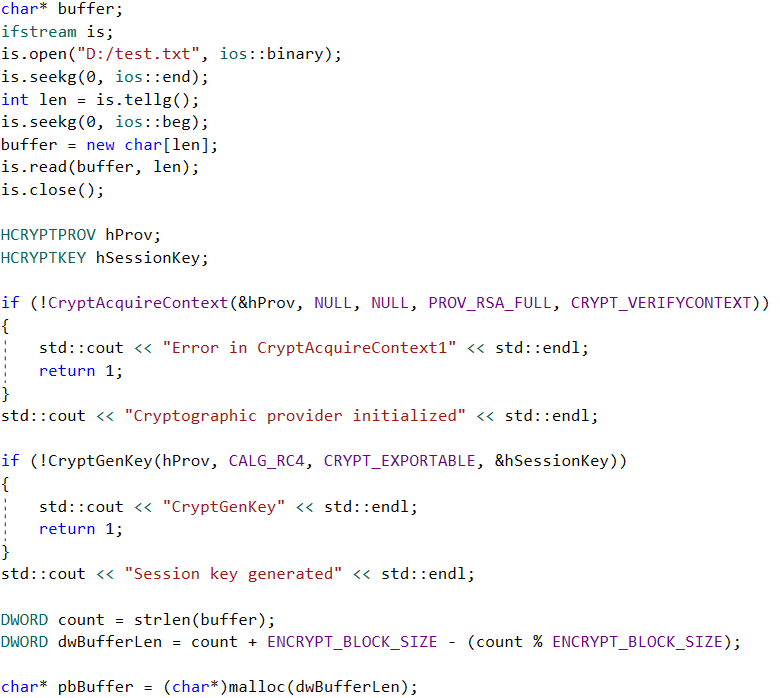
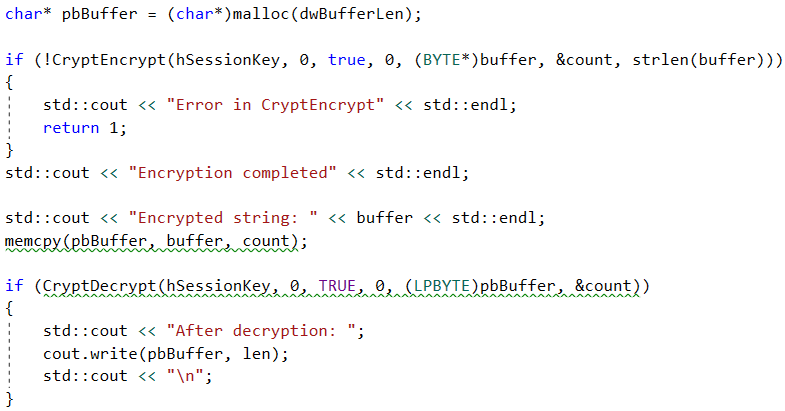
system("PAUSE");

return 0;

}

# **3 Задание**

1. Вы видите, что строка расшифрована не совсем корректно. Исходный вариант – Test. Следовательно, нужно из массива pbBuffer вытянуть только те байты, которые были в исходной строке.
2. Используя приведенную программу зашифруйте и расшифруйте текстовый файл.
3. Rsa, зашифровать картинку jpg

1. Нужно ввести пароль, который забивается при вводе звездочками.
2. Пароль нужно зашифровать, используя программу в части 1.
3. Зашифрованный пароль нужно сравнить с шифром, заранее помещенным в файл на диске. При совпадении шифров выдать ответ

You are welcome.

