МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ИЖЕВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. М.Т. Калашникова»

Кафедра «Защита информации в компьютеризированных системах»

Отчёт по лабораторной работе №7

по дисциплине «Программно-аппаратные средства обеспечения информационной безопасности»

на тему «Шифрование – дешифрование файлов с использованием стандартных функций CryptoApi»

Выполнил:

студент группы С08-361-1

Максимова А.В.

Проверил:

к.т.н, доцент

Вдовин А. Ю.

Ижевск 2017

Цель работы:

Изучение основных принципов шифрования-дешифрования файлов с использованием стандартных функций CryptoApi.

Задание:

Создание программы для шифрования-дешифрования файлов с использованием стандартных функций CryptoApi.

Листинг программы:

CCrypto.h

#pragma once

#include <iostream>

#include <Windows.h>

class CCrypto

{

public:

CCrypto();

bool CryptoEncryptFile(); // Шифрование файла

bool CryptoDecryptFile(); // Дешифрование файла

public:

char inputFile[255] = "D:\\Study\\8 семестр\\ПАСОИБ\\Лаба 3\\CriptoApi\\Debug\\1.txt"; // Исходный файл

char outputFile[255] = "D:\\Study\\8 семестр\\ПАСОИБ\\Лаба 3\\CriptoApi\\Debug\\2.txt"; // Выходной файл

private:

HCRYPTPROV hProv = NULL; // Криптопровайдер

HCRYPTKEY hKey = NULL; // Криптоключ

char\* szPassword = "";

DWORD dwPasswordLen = 0;

LPBYTE pEncryptedData = NULL;

DWORD dwKeyLen = 0, dwValLen = 0;

DWORD dwEncryptedDataLen = 0;

};

CCrypto.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#ifndef \_WIN32\_WINNT // Allow use of features specific to Windows 2000 or later.

#define \_WIN32\_WINNT 0x0500 // Change this to the appropriate value to target other versions of Windows.

#endif

#include <fstream>

#include "CCrypto.h"

#pragma comment(lib, "Crypt32")

const int MAX\_LEN = 10;

const int len = 128;

CCrypto::CCrypto()

{

//inputFile = "1.txt";

//outputFile = "2.txt";

}

bool CCrypto::CryptoEncryptFile()

{

bool result = false;

BOOL bStatus = FALSE;

#pragma region CryptAcquireContext

bStatus = CryptAcquireContext(&hProv,

NULL, // Контейнер по умолчанию

MS\_ENHANCED\_PROV,

PROV\_RSA\_FULL,

0);

if (!bStatus)

{

if (CryptAcquireContext(

&hProv,

NULL,

MS\_ENHANCED\_PROV,

PROV\_RSA\_FULL,

CRYPT\_NEWKEYSET))

{

printf(TEXT("CryptAcquireContext succeeded.\n"));

}

else

{

printf(TEXT("Could not create the default key container.\n"));

}

}

#pragma endregion

#pragma region CryptGetUserKey

bStatus = CryptGetUserKey(hProv,

AT\_KEYEXCHANGE,

&hKey);

if (!bStatus)

{

if (NTE\_NO\_KEY == GetLastError())

{

// Нет ключа. Сгенерируем ключ

if (!CryptGenKey(

hProv,

AT\_KEYEXCHANGE,

CRYPT\_EXPORTABLE,

&hKey))

{

printf("Could not create a user public key.\n");

return false;

}

}

else

{

printf("User public key is not available and may not exist.\n");

return false;

}

}

#pragma endregion

#pragma region CryptGetKeyParam

dwValLen = sizeof(DWORD); // Размер ключа

bStatus = CryptGetKeyParam(hKey,

KP\_KEYLEN,

(LPBYTE)&dwKeyLen,

&dwValLen,

0);

if (!bStatus)

{

printf("CryptGetKeyParam failed with error 0x%.8X\n", GetLastError());

return false;

}

#pragma endregion

#pragma region CryptEncrypt

//Выделим буффер

dwKeyLen = (dwKeyLen + 7) / 8; // Преобразуем в байты

char buffer[MAX\_LEN];

FILE \*f1, \*f2;

f1 = fopen(inputFile, "rb+");

f2 = fopen(outputFile, "wb+");

char buf;

while (!feof(f1))

{

pEncryptedData = (LPBYTE)LocalAlloc(0, dwKeyLen);

for (int i = 0; i < MAX\_LEN; ++i)

buffer[i] = '\0';

fread(&buffer, MAX\_LEN, 1, f1);

szPassword = &buffer[0];

dwPasswordLen = MAX\_LEN;

CopyMemory(pEncryptedData, szPassword, dwPasswordLen);

dwEncryptedDataLen = dwPasswordLen;

bStatus = CryptEncrypt(hKey,

NULL,

TRUE,

0,

pEncryptedData,

&dwEncryptedDataLen,

dwKeyLen);

if (!bStatus)

{

printf("CryptEncrypt failed with error 0x%.8X\n", GetLastError());

return false;

}

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

{

buf = pEncryptedData[i];

fwrite(&buf, sizeof(char), 1, f2);

}

}

fclose(f2);

fclose(f1);

#pragma endregion

return result;

}

bool CCrypto::CryptoDecryptFile()

{

BOOL bStatus = FALSE;

bool result = false;

char buffer[len];

FILE \*f;

char buf;

f = fopen(outputFile, "rb+");

while (!feof(f))

{

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

buffer[i] = '\0';

fread(&buffer, len, 1, f);

szPassword = &buffer[0];

dwPasswordLen = len;

dwEncryptedDataLen = len;

pEncryptedData = NULL;

pEncryptedData = (LPBYTE)LocalAlloc(0, dwKeyLen);

CopyMemory(pEncryptedData, szPassword, dwPasswordLen);

bStatus = CryptDecrypt(hKey,

NULL,

TRUE,

0,

pEncryptedData,

&dwEncryptedDataLen);

for (DWORD i = 0; i < MAX\_LEN; i++)

printf("%c", pEncryptedData[i]);

}

fclose(f);

return result;

}

Main.cpp

#include <stdio.h>

#include <tchar.h>

#include <windows.h>

#include <iostream>

#include "CCrypto.h"

int main()

{

CCrypto crypto;

crypto.CryptoEncryptFile();

crypto.CryptoDecryptFile();

std::cout << std::endl;

system("pause");

}

Скриншоты работы программы:

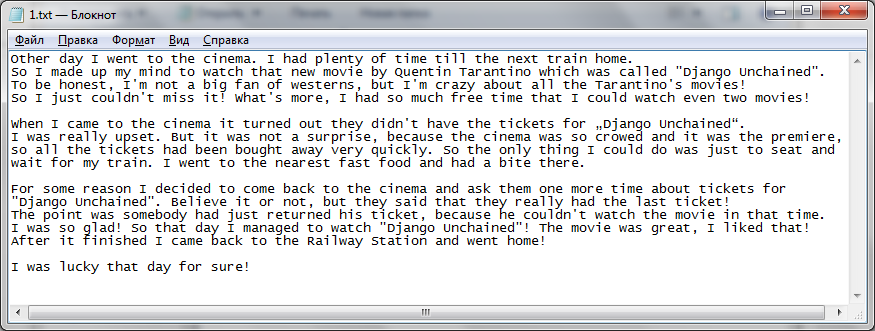


Рис. 1 – Содержимое исходного файла

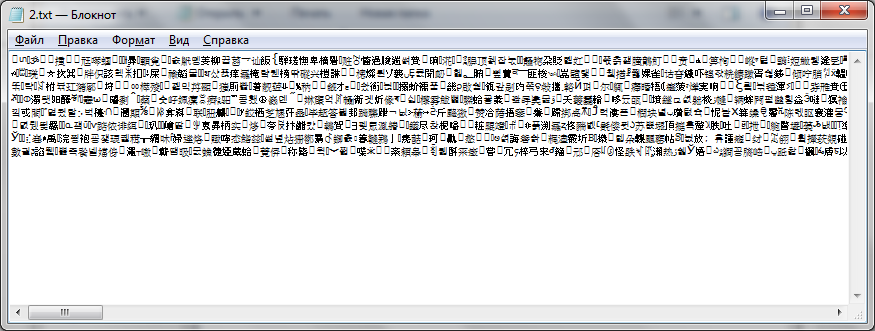


Рис. 2 – Содержимое зашифрованного файла

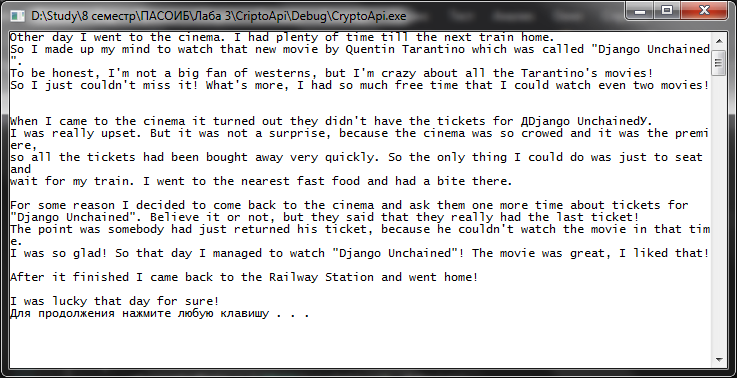


Рис. 3 – Результат работы программы

Вывод:

В данной лабораторной работе были изучены основные принципы шифрования-дешифрования файлов с использованием стандартных функций CryptoApi. Шифрование происходило алгоритмом RSA.