|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
|  | | |
| Практическое задание № 4 | | |
| по дисциплине «Основы криптографии» | | |
|  | | |
| **.** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМ-71 |
| Бригада: | 24 |
| Студенты: | Востриков Вячеслав, |
|  | Баштовой Павел |
|  |  |
| Преподаватели: | Ступаков Илья Михайлович |
|  |  |
|  | | |
| Новосибирск | | |
| 2019 | | |

**Цель**

Научится использовать готовые криптографические примитивы для шифрования данных.

**Ход работы**

Сделать программу, которая шифрует и дешифрует некоторый файл с помощью алгоритма AES. В качестве ключа использовать хеш (с возможностью выбора алгоритма) от вводимого пользователем пароля. Сам ключ в итоге нигде сохраняться не должен. Использовать режим CBC и в качестве IV взять ключ (такой подход считается плохим, подумайте почему).

**Текст программы**

using System;

using System.IO;

using System.Security.Cryptography;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_4

{

class Program

{

public static byte[] AES\_Cipher(byte[] password\_hash, byte[] data)

{

password\_hash = password\_hash.Where((item, i) => i < 16).ToArray();

Aes crypt\_AES = Aes.Create();

crypt\_AES.IV = password\_hash;

crypt\_AES.Key = password\_hash;

crypt\_AES.Mode = CipherMode.CBC;

crypt\_AES.Padding = PaddingMode.Zeros;

byte[] encrypted;

using (ICryptoTransform crypt = crypt\_AES.CreateEncryptor(crypt\_AES.Key, crypt\_AES.IV))

{

using (MemoryStream ms = new MemoryStream())

{

using (CryptoStream cs = new CryptoStream(ms, crypt, CryptoStreamMode.Write))

{

cs.Write(data, 0, data.Length);

}

encrypted = ms.ToArray();

}

}

return encrypted;

}

public static byte[] AES\_Decipher(byte[] password\_hash2, byte[] cip\_data)

{

password\_hash2 = password\_hash2.Where((item, i) => i < 16).ToArray();

var crypt\_AES = Aes.Create();

crypt\_AES.Key = password\_hash2;

crypt\_AES.IV = password\_hash2;

crypt\_AES.Mode = CipherMode.CBC;

crypt\_AES.Padding = PaddingMode.Zeros;

byte[] data;

using (ICryptoTransform crypt = crypt\_AES.CreateDecryptor(crypt\_AES.Key, crypt\_AES.IV))

{

using (MemoryStream ms = new MemoryStream())

{

using (CryptoStream cs = new CryptoStream(ms, crypt, CryptoStreamMode.Write))

{

cs.Write(cip\_data, 0, cip\_data.Length);

}

data = ms.ToArray();

}

}

return data;

}

static void Main(string[] args)

{

int choice = 0;

Console.WriteLine("1 - Шифровать файл, 2 - Дешифровка файла.");

choice = Console.Read();

Console.ReadLine();

if (choice == '1') // Шифрование

{

Console.WriteLine("Введите пароль: ");

string password = Console.ReadLine();

byte[] password\_byte = Encoding.Default.GetBytes(password);

Console.WriteLine("Выберите алгоритм для хэширования. 1 - SHA256, 2 - SHA1, 3 - MD5");

int p = Console.Read();

Console.ReadLine();

bool flag = true;

byte[] password\_hash = new byte[32];

switch (p)

{

case '1':

{

var hash\_sha256 = HashAlgorithm.Create("SHA256");

password\_hash = hash\_sha256.ComputeHash(password\_byte);

break;

}

case '2':

{

var hash\_sha1 = HashAlgorithm.Create("SHA1");

password\_hash = hash\_sha1.ComputeHash(password\_byte);

break;

}

case '3':

{

var hash\_md5 = HashAlgorithm.Create("MD5");

password\_hash = hash\_md5.ComputeHash(password\_byte);

break;

}

default:

flag = false;

break;

}

if (flag)

{

Console.WriteLine("Введите имя оригинального файла!");

var string1 = Console.ReadLine();

var bytes\_fr\_image = File.ReadAllBytes(string1);

var crp\_data = AES\_Cipher(password\_hash, bytes\_fr\_image);

Console.WriteLine("Введите имя для результата шифрования!");

var string2 = Console.ReadLine();

File.WriteAllBytes(string2, crp\_data);

}

}

else if (choice == '2') // Дешифровка

{

Console.WriteLine("Введите пароль: ");

string password2 = Console.ReadLine();

byte[] password\_byte2 = Encoding.Default.GetBytes(password2);

Console.WriteLine("Выберите алгоритм для хэширования. 1 - SHA256, 2 - SHA1, 3 - MD5");

int p = Console.Read();

Console.ReadLine();

byte[] password\_hash2 = new byte[32];

bool flag = true;

switch (p)

{

case '1':

{

var hash\_sha256 = HashAlgorithm.Create("SHA256");

password\_hash2 = hash\_sha256.ComputeHash(password\_byte2);

break;

}

case '2':

{

var hash\_sha1 = HashAlgorithm.Create("SHA1");

password\_hash2 = hash\_sha1.ComputeHash(password\_byte2);

break;

}

case '3':

{

var hash\_md5 = HashAlgorithm.Create("MD5");

password\_hash2 = hash\_md5.ComputeHash(password\_byte2);

break;

}

default:

flag = false;

break;

}

if (flag)

{

Console.WriteLine("Введите имя шифрованного файла!");

var string1 = Console.ReadLine();

var bytes\_fr\_image = File.ReadAllBytes(string1);

var decr\_data = AES\_Decipher(password\_hash2, bytes\_fr\_image);

Console.WriteLine("Введите имя для записи результата расшифровки!");

var string2 = Console.ReadLine();

File.WriteAllBytes(string2, decr\_data);

}

}

else

Console.WriteLine("Некорректный выбор! Повторите снова.");

}

}

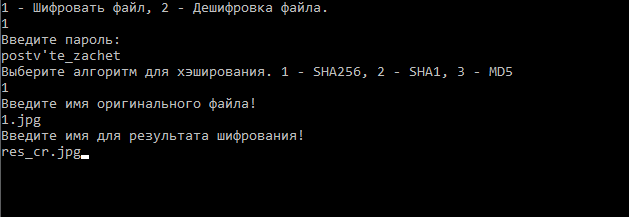
}

Сделать селфи бригады (в полном составе, должно быть видно лица), зашифровать и выложить на общий диск. Пароль указать в отчете.

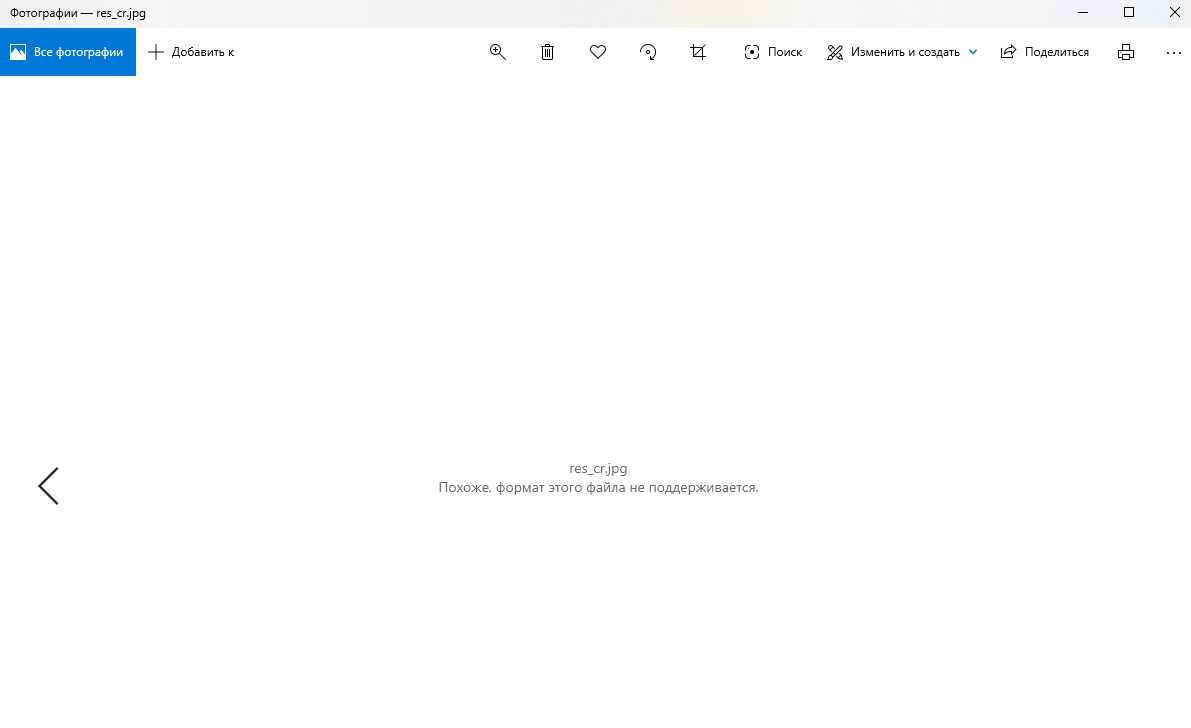
Файл 1.jpg. Пароль: postv’te\_zachet



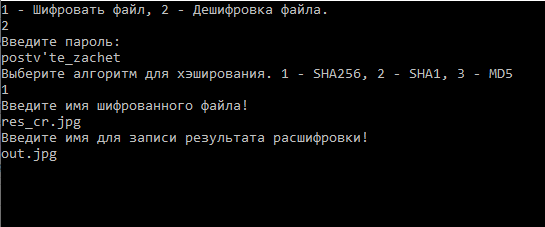
Шифрование картинки:



Результат шифрования res\_cr.jpg



Расшифровка картинки:



Результат расшифровки out.jpg

