|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
| C:\Users\Дарья\Desktop\3 ЦВЕТ.png | | |
| Кафедра прикладной математики | | |
|  | | |
| Практическая работа № 4 | | |
| по дисциплине «Основы криптографии» | | |
|  | | |
|  | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМ-72 |
| Бригада: | 3 |
| Студенты: | Голованова Дарья, |
|  | Журавлёва Юлия, |
| Преподаватель: | Ступаков Илья Михайлович |
|  |  |
|  |  |
|  | | |
| Новосибирск | | |
| 2019 | | |

**1. Цель работы**

Научится использовать готовые криптографические примитивы для шифрования данных.

**2. Порядок выполнения работы**

**Задание:**

* Написать программу, которая шифрует и дешифрует некоторый файл с помощью алгоритма AES:

Использовать хеш (с возможностью выбора алгоритма SHA-256, SHA1 или MD5) от вводимого пользователем пароля в качестве ключа (не сохраняя его). Использовать режим CBC, в качестве IV взять ключ.

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

**Код 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 OK\_4

{

class Program

{

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

{

password\_hash = password\_hash.Where((item, i) => i < 16).ToArray(); //Фильтрация последовательности значений на основе заданного предиката

Aes crypt\_AES = Aes.Create(); //Абстрактный базовый класс, от которого наследуются все реализации стандарта AES

crypt\_AES.IV = password\_hash; //Абстрактный базовый класс c ключом, вектором инициализации IV в режиме сцепления блоков шифротекста CBC,типом заполнения Zeros

crypt\_AES.Key = password\_hash;

crypt\_AES.Mode = CipherMode.CBC;

crypt\_AES.Padding = PaddingMode.Zeros;

byte[] cipher;

//Объект-шифратор для алгоритма симметричного шифрования с текущим свойством Key и вектором инициализации IV

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

{

using (MemoryStream ms = new MemoryStream()) //Создаем поток, память которого - резервное хранилище

{

using (CryptoStream cs = new CryptoStream(ms, ict, CryptoStreamMode.Write)) //Поток, связывающий потоки данных с криптографическими преобразованиями

cs.Write(data, 0, data.Length); //Текстовое представление заданного значения в стандартный выходной поток

cipher = ms.ToArray();

}

}

return cipher;

}

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

{

password\_hash2 = password\_hash2.Where((item, i) => i < 16).ToArray(); //Фильтрация последовательности значений на основе заданного предиката

var crypt\_AES = Aes.Create(); //Абстрактный базовый класс, от которого наследуются все реализации стандарта AES

crypt\_AES.Key = password\_hash2; //Абстрактный базовый класс c ключом, вектором инициализации IV в режиме сцепления блоков шифротекста CBC,типом заполнения Zeros

crypt\_AES.IV = password\_hash2;

crypt\_AES.Mode = CipherMode.CBC;

crypt\_AES.Padding = PaddingMode.Zeros;

byte[] file ;

//Объект - шифратор для алгоритма симметричного шифрования с текущим свойством Key и вектором инициализации IV

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

{

using (MemoryStream ms = new MemoryStream()) //Создаем поток, память которого резервное хранилище

{

using (CryptoStream cs = new CryptoStream(ms, ict, CryptoStreamMode.Write)) //Поток, связывающий потоки данных с криптографическими преобразованиями

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

file = ms.ToArray();

}

}

return file;

}

static void Main(string[] args)

{

int choice;

Console.WriteLine("1 - Зашифровать файл, 2 - Расшифровать файл");

choice = Console.Read();

Console.ReadLine();

if (choice == '1') // Шифрование

{

Console.WriteLine("Введите пароль: ");

byte[] password\_byte = Encoding.Default.GetBytes(Console.ReadLine()); //Запись пароля в массив байтов

Console.WriteLine("SHA256, SHA1 или MD5: ");

string alg = Console.ReadLine();

int length\_of\_h;

if (alg == "SHA256")

length\_of\_h = 32;

else

if (alg == "SHA1")

length\_of\_h = 20;

else

if (alg == "MD5")

length\_of\_h = 16;

else

length\_of\_h = 0;

byte[] password\_hash = new byte[length\_of\_h];

var hash = HashAlgorithm.Create(alg); //Создает экземпляр реализации хэш-алгоритма

password\_hash = hash.ComputeHash(password\_byte); //Вычисляет хэш-значение для входных данных

if (length\_of\_h != 0)

{

Console.WriteLine("Введите имя файла: ");

var bytes\_fr\_image = File.ReadAllBytes(Console.ReadLine()); //Запись содержимого файла в массив байтов

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

Console.WriteLine("Введите имя файла для результата: ");

File.WriteAllBytes(Console.ReadLine(), crp\_data); //Запись в файл указанного массив байтов

}

}

else if (choice == '2') // Дешифровка

{

Console.WriteLine("Введите пароль: ");

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

Console.WriteLine("SHA256, SHA1 или MD5: ");

string alg = Console.ReadLine();

int length\_of\_h;

if (alg == "SHA256")

length\_of\_h = 32;

else

if (alg == "SHA1")

length\_of\_h = 20;

else

if (alg == "MD5")

length\_of\_h = 16;

else

length\_of\_h = 0;

byte[] password\_hash2 = new byte[length\_of\_h];

var hash = HashAlgorithm.Create(alg); //Создает экземпляр реализации хэш-алгоритма

password\_hash2 = hash.ComputeHash(password\_byte2); //Вычисляет хэш-значение для входных данных

if (length\_of\_h != 0)

{

Console.WriteLine("Введите имя зашифрованного файла: ");

var bytes\_fr\_image = File.ReadAllBytes(Console.ReadLine()); //Запись содержимого файла в массив байтов

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

Console.WriteLine("Введите имя файла для результата: ");

File.WriteAllBytes(Console.ReadLine(), decr\_data); //Запись в файл указанного массив байтов

}

}

else Console.WriteLine("Ошибка ввода!");

}

}

}

**Фото бригады:**

****

**Зашифровка фотографии:**

Пароль: aminstu

