Государственное образовательное учреждение

«Белорусский государственный технологический университет»

Отчет

*по лабораторной работе №12*

ТЕМА ЗАНЯТИЯ: ИЗУЧЕНИЕ СТАНДАРТНЫХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ ПРИЛОЖЕНИЙ, ИСПОЛЬЗУЮЩИХ СИММЕТРИЧНОЕ И АССИМЕТРИЧНОЕ ШИФРОВАНИЕ, С ПРИМЕНЕНИЕМ БИБЛИОТЕКИ SYSTEM.SECURITY.CRYPTOGRAPHY

|  |  |
| --- | --- |
|  | **Исполнитель:**  Студента (-ка) группы 2  Специальность\_\_\_\_ИСиТ\_\_\_  Гулевич Никита Максимович  (Ф.И.О.) |
|  | **Преподаватель**:  Ржеутская Н. В. |

Минск, 2025

**Цель**: изучить модель криптографии .NET Framework, основные классы и структуры данных, разработать приложение для шифрования файлов, использующих симметричные и ассиметричные алгоритмы шифрования.

using System;

using System.IO;

using System.Security.Cryptography;

using System.Text;

class CryptoProgram

{

static void Main()

{

string surname = "Гулевич";

// Генерация ключа и IV для TripleDES

using (TripleDES tripleDes = TripleDES.Create())

{

tripleDes.KeySize = 192;

tripleDes.GenerateKey();

tripleDes.GenerateIV();

// Шифрование

byte[] encryptedData = EncryptString(surname, tripleDes.Key, tripleDes.IV);

// Дешифрование

string decryptedData = DecryptString(encryptedData, tripleDes.Key, tripleDes.IV);

// Хеширование SHA384

byte[] hash = ComputeSHA384Hash(surname);

// Сохранение результатов в файлы

SaveToFile("key.bin", tripleDes.Key);

SaveToFile("iv.bin", tripleDes.IV);

SaveToFile("encrypted.bin", encryptedData);

SaveToFile("hash.bin", hash);

Console.WriteLine("Исходная фамилия: " + surname);

Console.WriteLine("Зашифрованные данные (hex): " + BitConverter.ToString(encryptedData).Replace("-", ""));

Console.WriteLine("Расшифрованные данные: " + decryptedData);

Console.WriteLine("Хеш SHA384 (hex): " + BitConverter.ToString(hash).Replace("-", ""));

// Проверка ЭЦП-подобной верификации

VerifyHash(surname, hash, "Оригинальная");

// Демонстрация изменения сообщения

VerifyHash("Гулевич1", hash, "Измененное сообщение");

// Демонстрация изменения хеша

byte[] modifiedHash = (byte[])hash.Clone();

modifiedHash[0] ^= 0xFF; // Инвертируем первый байт

VerifyHash(surname, modifiedHash, "Измененный хеш");

}

}

static byte[] EncryptString(string plainText, byte[] key, byte[] iv)

{

using (TripleDES tripleDes = TripleDES.Create())

{

tripleDes.Key = key;

tripleDes.IV = iv;

using (var encryptor = tripleDes.CreateEncryptor())

using (var ms = new MemoryStream())

{

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

using (var sw = new StreamWriter(cs))

{

sw.Write(plainText);

}

return ms.ToArray();

}

}

}

static string DecryptString(byte[] cipherText, byte[] key, byte[] iv)

{

using (TripleDES tripleDes = TripleDES.Create())

{

tripleDes.Key = key;

tripleDes.IV = iv;

using (var decryptor = tripleDes.CreateDecryptor())

using (var ms = new MemoryStream(cipherText))

using (var cs = new CryptoStream(ms, decryptor, CryptoStreamMode.Read))

using (var sr = new StreamReader(cs))

{

return sr.ReadToEnd();

}

}

}

static byte[] ComputeSHA384Hash(string input)

{

using (SHA384 sha384 = SHA384.Create())

{

return sha384.ComputeHash(Encoding.UTF8.GetBytes(input));

}

}

static void SaveToFile(string filename, byte[] data)

{

File.WriteAllBytes(filename, data);

}

static void VerifyHash(string message, byte[] hash, string testName)

{

byte[] computedHash = ComputeSHA384Hash(message);

bool isValid = hash.Length == computedHash.Length &&

hash.SequenceEqual(computedHash);

Console.WriteLine($"\nПроверка {testName}:");

Console.WriteLine($"Результат верификации: {(isValid ? "Действителен" : "Недействителен")}");

}

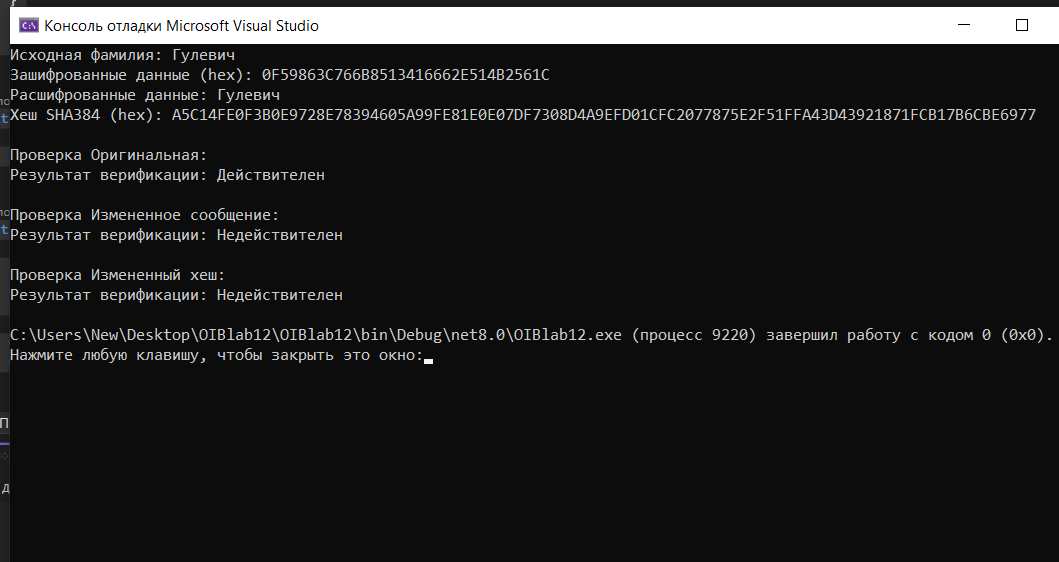
}

1. Ознакомьтесь с созданием криптографического приложения.

2. Выполните шифрование, дешифрование и хеширование своей фамилии по указанным алгоритмам. Используя функции работы с файлами, сохраните ключи шифрования, результаты шифрования и хеширования.

3. Для выше указанных алгоритмов, используя Hex-редактор, продемонстрируйте ключи шифрования, зашифрованные и захешированные данные.

4. Реализуйте проверку сообщения (фамилии) и хеша по примеру ЭЦП. Продемонстрируйте, что произойдет, если будет изменен хеш илисообщение. Задания того, чтонужнореализовать, представленыниже.



**Список использованных источников**

1. Ржеутская Н. В., Нистюк О. А., Уласевич Н. И. Основы защиты информации: лабораторный практикум. Минск: Учреждение образования «Белорусский государственный технологический университет», 2024.