Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра ПМиК

**Лабораторная работа №4**

«Криптографические хеш-функции»

Выполнил:

студент гр. МГ-211 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Бурдуковский И.А./

подпись

Проверил:

Профессор

кафедры ПМиК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Фионов А.Н./

Новосибирск

2023 г.

**ОГЛАВЛЕНИЕ**

[Задание 3](#_Toc135774551)

[Выполнение 4](#_Toc135774552)

[Листинг 5](#_Toc135774553)

# Задание

1. Выбрать одну понравившуюся хеш-функцию и интегрировать ее в приложение для передачи файла в зашифрованном виде по сети, как это требуют протоколы выработки секретного ключа. Сравнить время вычисления хеш-функции с временем выполнения других операций в протоколе.

# Выполнение

Была выбрана хеш-функция SHA-3, а в качестве блокового шифра HC-128.

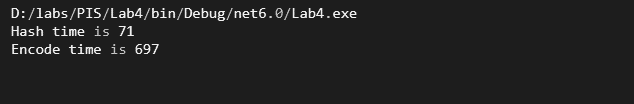
Замеры времени для хеш-функции были сделаны через функцию System.Net.DateTime.

Так же через эти сокеты перед передачей файла происходит инициализация DH MQV и затем шифрование файла блоковым шифром.

После, значение этой программы будем использовать как ключ для кодирования файла.

По итогу время хеширования занимает несколько больше времени в сравнении с кодированием блочным шифром HC-128 на малых объемах блока. Но на большем количестве данных шифрование уже значительно превышает время чем на вычисления хеша от ключа.

К тому же его стойкость возрастает при использовании хеш-функции для секретного ключа.



# Листинг

var t1\_s = DateTime.Now;

var processGetSha3 = new Process

{

StartInfo = new ProcessStartInfo

{

FileName = "D:\\labs\\PIS\\Lab4\\x64\\Debug\\Lab4.exe",

// DATA SIZE

Arguments = $"{mqv\_key} 16",

UseShellExecute = false,

RedirectStandardOutput = true,

RedirectStandardInput = true,

RedirectStandardError = true,

CreateNoWindow = true

}

};

if (!processGetSha3.Start())

throw new ApplicationException();

string mqv\_key\_hashed = null;

while (!processGetSha3.StandardOutput.EndOfStream)

{

mqv\_key\_hashed = processGetSha3.StandardOutput.ReadLine();

}

processGetSha3.WaitForExit();

Console.WriteLine($"Hash time: {(DateTime.Now - t1\_s).Milliseconds}");

var t2\_s = DateTime.Now;

var processHC128 = new Process

{

StartInfo = new ProcessStartInfo

{

FileName = "E:\\labs\\PIS\\Lab4\\x64\\Debug\\Lab4.exe",

// K IV

Arguments = $"{mqv\_key\_hashed.Substring(0,32)} 00000000000000000000000000000000",

UseShellExecute = false,

RedirectStandardOutput = true,

RedirectStandardInput = true,

RedirectStandardError = true,

CreateNoWindow = true

}

};

if (!processHC128.Start())

throw new ApplicationException();

string encoded\_data = null;

while (!processHC128.StandardOutput.EndOfStream)

{

encoded\_data = processHC128.StandardOutput.ReadLine();

}

processHC128.WaitForExit();

Console.WriteLine($"Encode time: {(DateTime.Now - t2\_s).Milliseconds}");