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

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.

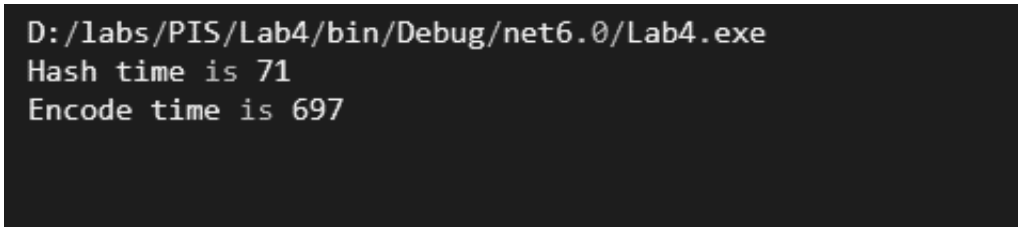
Замеры времени для хеш-функции были сделаны через функцию System.Net.DateTime.

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}");
```