\*МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №11 НА ТЕМУ:**

**Исследование криптографических хеш-функций**

                                                                 Выполнил студент 3 курса 5 группы

Дмитрук Илья Игоревич

Минск 2024

# Задание 1.

Хеш-функция – математическая или иная функция *h* = *H(М)*, которая принимает на входе строку символов *М*, называемую также прообразом, переменной длины n и преобразует ее в выходную строку фиксированной (обычно – меньшей) длины *l*.

Хеширование (или хэширование, англ. hashing) – это преобразование входного массива данных определенного типа и произвольной длины (практически) в выходную битовую строку фиксированной длины.

Входное сообщение «дополняется» (расширяется) так, чтобы его длина (в битах) была конгруэнтной к 448 по модулю 512. Эта операция выполняется следующим образом: один бит «1» добавляется к сообщению, а затем добавляются биты «0», так что длина в битах дополненного сообщения стала конгруэнтной 448 по модулю 512. Добавляется не менее одного бита, но не более 448 битов.

Алгоритм хеширования SHA-256 работает следующим образом: сначала, с помощью восьми 32-битных значений, инициализируются внутренние регистры. Затем сообщение разбивается на блоки длинной 512 бит. Затем каждый блок проходит через 64 раунда (итерации) вычислений. На каждом раунде используются дополнительные константы и временные переменные. Внутренние регистры обновляются с использованием различных побитовых операций (логические операции, сложение, циклические сдвиги и т. д.).

Так же к исходному сообщению, перед хешированием, добавляется соль. Соль ­– это случайная строка данных, которая добавляется к паролю или другому входному значению перед хешированием. Она обеспечивает уникальность хеша (даже если хешируются два одинаковых сообщения). Алгоритм генерации соли реализован в функции GenerateSalt. Её код представлен в листинге 1.1. В листинге 1.2 представлен код функции HashSHA256, которая выполняет хеширование алгоритмом SHA-256.

public static string GenerateSalt(int length)

{

byte[] saltBytes = new byte[length];

using (var rng = new RNGCryptoServiceProvider())

{

rng.GetBytes(saltBytes);

}

return Convert.ToBase64String(saltBytes);

}

Листинг 1.1 – Функция генерации соли

public static string HashSHA256(string rawData)

{

using (SHA256 sha256Hash = SHA256.Create())

{

byte[] bytes = sha256Hash.ComputeHash(Encoding.UTF8.GetBytes(rawData));

StringBuilder builder = new StringBuilder();

for (int i = 0; i < bytes.Length; i++)

{

builder.Append(bytes[i].ToString("x2"));

}

return builder.ToString();

}

}

Листинг 1.2 – Функция хеширования алгоритмом SHA-256

Результат выполнения хеширования представлен на рисунке 1.1.

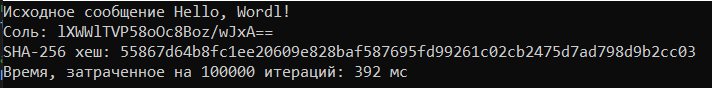


Рисунок 1.1 – Результат хеширования алгоритмом SHA-256

# Задание 2.

На рисунке 2.1 представлен график зависимости времени работы алгоритма SHA-256 от количества итераций выполнения алгоритма.

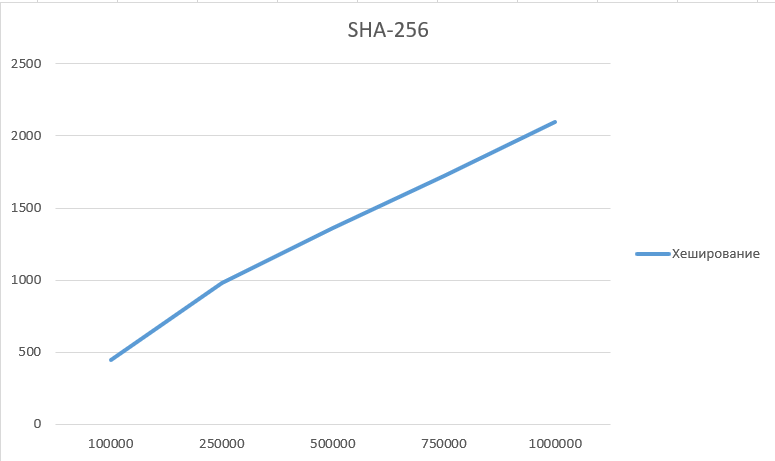


Рисунок 2.1 – Время хеширования алгоритмом SHA-256

**Вывод**

В ходе лабораторной работы были изучены различные способы хеширования. Было разработано приложение, выполняющее алгоритм хеширования SHA-256. Так же была измерена скорость выполнения данного алгоритма.