Министерство образования Республики Беларусь

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Отчет к лабораторной работе № 8

«Исследование криптографических хеш-функций»

Выполнила:

студентка 3 курса 2 группы

Шастовская М. С.

Вариант: 12

Преподаватель:

Сазонова Д.В.

Минск 2023

**Цель:** изучение алгоритмов хеширования и приобретение практических навыков их реализации и использования в криптографии.

**Задачи:**

* Закрепить теоретические знания по алгебраическому описанию и алгоритмам реализации операций вычисления однонаправленных хеш-функций.
* Освоить методику оценки криптостойкости хеш-преобразований на основе «парадокса дня рождения».
* Разработать приложение для реализации заданного алгоритма хеширования (из семейств MD и SHA).
* Оценить скорость вычисления кодов хеш-функций.
* Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента. **1 Теоретические сведения**

*Определение 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.

*Определение 2*. Хеширование (или хэширование, англ. hashing) – это преобразование входного массива данных определенного типа и произвольной длины (практически) в выходную битовую строку фиксированной длины.

*Определение 3.* Криптографическая хеш-функция – это специальный класс хеш-функций, который имеет различные свойства, необходимые для решения задач в области криптографии.

Основные задачи, решаемые с помощью хеш-функций:

* аутентификация (хранение паролей);
* проверка целостности данных;
* защита файлов;
* обнаружение зловредного ПО;
* криптовалютные технологии.

*Свойство 1.* Детерминированность: независимо от того, сколько раз вычисляется H(M), M – const, при использовании одинакового алгоритма код хеш-преобразования h всегда должен быть одинаковым.

*Свойство 2.* Скорость вычисления хеша h: если процесс вычисления h недостаточно быстрый, система просто не будет эффективной.

*Свойство 3.* Сложность обратного вычисления: для известного H(М) невозможно (практически) определить М. Это важнейшее свойство хеш-функции для криптографических применений – свойство односторонности преобразования.

Мерой криптостойкости хеш-функции считается вычислительная сложность нахождения коллизии. Для хеш-функций одним из основных средств поиска коллизий является метод, основанный на известной статистической задаче – «парадоксе дня рождения».

Парадоксом является высокая (как кажется на первый взгляд) вероятность наступления указанного события. При этом предполагается, что:

* в этой группе нет близнецов;
* люди рождаются независимо друг от друга, т. е. дата (день) рождения любого человека не влияет на дату рождения другого;
* люди рождаются равномерно и случайно, т. е. люди с равной вероятностью могут рождаться в любой день года; с формальной точки зрения это означает, что вероятность р1 рождения отдельно выбранного члена группы (как и любого человека) в любой выбранный день равна р1 = 1 / 365 (хотя известно, что в реальности рождение людей не совсем соответствует такому предположению).

Определение 5. Хеш-функция – это функция, выполняющая отображение из множества М в число, находящееся в интервале [0, m – 1]: h: M → [0, m – 1].

Базовые алгоритмы обоих рассматриваемых семейств (MD и SHA) условно можно разделить на 5 стадий:

* расширение входного сообщения;
* разбивка расширенного сообщения на блоки;
* инициализация начальных констант;
* обработка сообщения поблочно (основная процедура алгоритма хеширования);
* вывод результата

Входное сообщение «дополняется» (расширяется) так, чтобы его длина (в битах) была конгруэнтной к 448 по модулю 512. Это значит, что сообщение начальной длиной L битов расширяется так, что остаются незаполненными всего лишь 64 бита, чтобы итоговая длина L' была кратной 512. В указанные 64 бита записывается двоичная длина.

Расширение происходит всегда, даже если длина сообщения уже соответствует 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.

2 Практическая часть

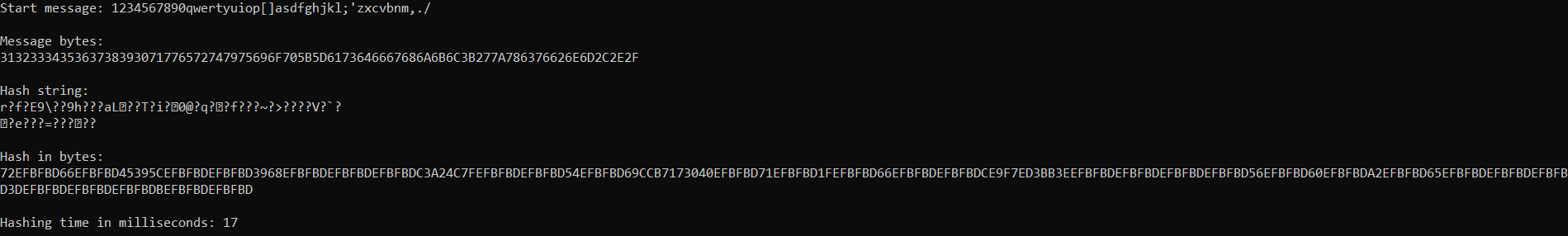


Рисунок 2.1 – Результат хэширования алгоритмом SHA-512.

На данных скриншотах отображен процесс работы программы хэширования алгоритмом семейства SHA.

Основные функции программы представлены на рисунке 2.2.

static void Main(string[] args)

{

Stopwatch sw = new Stopwatch();

string message = "1234567890qwertyuiop[]asdfghjkl;'zxcvbnm,./";

//for(int i = 0; i < 7; i++)

//{

// message += message;

//}

Console.WriteLine($"Start message: {message}");

sw.Start();

Console.WriteLine("\nMessage bytes:");

foreach (var item in Encoding.UTF8.GetBytes(message))

{

Console.Write(item.ToString("X"));

}

Console.WriteLine();

SHA512 shaM = new SHA512Managed();

message = Encoding.UTF8.GetString(shaM.ComputeHash(Encoding.UTF8.GetBytes(message)));

Console.WriteLine("\nHash string:");

Console.WriteLine(message);

Console.WriteLine("\nHash in bytes:");

foreach (var item in Encoding.UTF8.GetBytes(message))

{

Console.Write(item.ToString("X"));

}

sw.Stop();

Console.WriteLine($"\n\nHashing time in milliseconds: {sw.ElapsedMilliseconds}\n");

}

Рисунок 2.2 – Основные функции реализации.

Вывод

В ходе выполнения лабораторной работы были изучены и приобретены практические навыки разработки и использования приложений для реализации алгоритма семейства SHA.