Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Криптографические методы защиты информации

Студент: Сенченя В.И.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М. Г.

Минск 2023

**Лабораторная работа №11**

**Тема «****Исследование криптографических хеш-функций»**

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

Задачи:

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

**Хеширование**

**Хэширование** – это процесс преобразования произвольной входной последовательности (например, текстовой строки или битового потока) в фиксированный выходной хэш-код фиксированной длины. Хэш-код представляет собой результат вычисления хэш-функции, которая применяется к входным данным. Хэш-функции обладают свойством уникальности: для разных входных данных они генерируют разные хэш-коды. Однако, для одинаковых входных данных, хэш-функция всегда генерирует один и тот же хэш-код.

**SHA256**

**SHA256** – это один из наиболее распространенных криптографических хеш-алгоритмов. Алгоритм SHA256 принимает на вход сообщение длиной до 264-1 бит (включая биты длины сообщения) и выдает на выходе хеш-код длиной 256 бит. Хеш-код можно представить в виде последовательности из 64 шестнадцатеричных цифр (от 0 до F), каждая из которых кодирует 4 бита.

Процесс вычисления хеш-кода SHA256 можно разбить на несколько шагов:

1. Инициализация вектора начальных значений и таблицы констант.
2. Разбиение сообщения на блоки по 512 бит.
3. Дополнение последнего блока, если его длина меньше 512 бит.
4. Преобразование каждого блока через серию раундовых функций, каждый из которых использует нелинейную функцию, операции битовых сдвигов, XOR и AND.
5. Объединение результата каждого блока в итоговый хеш-код

Пример кода предоставлен на рисунке 1.1, результат работы на рисунке 1.2.

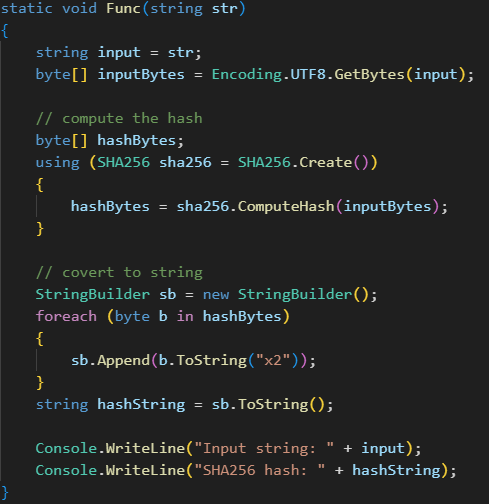


Рисунок 1.1 – Пример кода SHA256



Рисунок 1.2 – Результат работы хэш-функции

Скорость работы алгоритма предоставлена на рисунке 1.3.

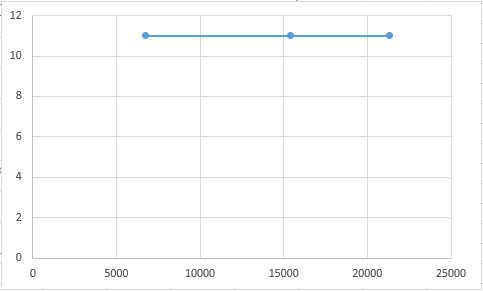


Рисунок 1.3 – Скорость выполнения алгоритма

**Вывод**

В ходе выполнения лабораторной работы я приобрёл и закрепил навыки практические навыки разработки приложения для реализации алгоритмов хеширования.