Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование криптографических

хэш функций

Студент: Димитриади А.В.

ФИТ 3 курс 5 группа

Преподаватель:

Савельева Маргарита Геннадьевна

Минск 2023

## 1. Описание приложения

Приложение позволяет выполнить следующие задачи:

* реализация алгоритма хэширования MD5;
* обработка входного сообщения, длина которых определяется спецификацией на данном алгоритме;
* оценка быстродействия данного алгоритма хеширования.

## 2. Методика выполнения поставленных задач

Алгоритм хэширования MD5 (Message Digest 5) преобразует входные данные произвольной длины в фиксированный хэш-код длиной 128 бит (16 байт). Вот базовый алгоритм MD5:

1. Инициализация переменных.

* Создание четырех 32-битных регистров (*A*, *B*, *C*, *D*), которые будут использоваться в процессе хэширования.
* Инициализация значений регистров начальными значениями, определенными спецификацией алгоритма.

1. Предварительная обработка (Padding) входных данных.

* Добавление бита 1 после входных данных.
* Добавление битов 0 до тех пор, пока длина данных не станет кратной 512 битам (64 байта).
* Добавление 64-битного блока, представляющего длину исходных данных в битах.

1. Разделение данных на блоки.

Разделение входных данных на блоки по 512 бит (64 байта).

1. Итерационное хэширование каждого блока.

* Для каждого блока данных выполнение четырех раундов.
* В каждом раунде проход по функциям, изменяющим значения регистров, и обработка текущего блока данных.

1. Формирование хэш-кода.

* Конкатенация значений регистров (*A*, *B*, *C*, *D*) в правильном порядке.
* Перевод каждого 32-битного регистра в 8-битный формат (16-ричное представление), чтобы получить окончательный хэш-код.

Алгоритм MD5 является криптографическим хэш-функцией, однако известны некоторые уязвимости и ограничения. В настоящее время рекомендуется использовать более безопасные алгоритмы хэширования, такие как SHA-256 или SHA-3.

Для реализации данного алгоритма использовалась библиотека «System.Security.Cryptography», алгоритм хэширования текста и отображение хэша представлен на рисунке 2.1.

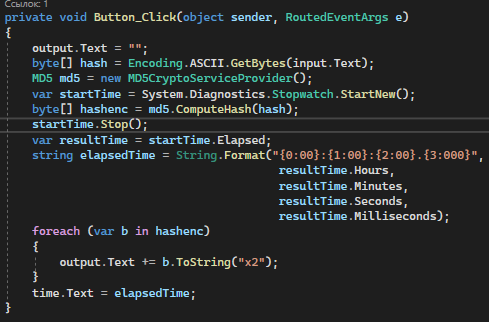


Рисунок 2.1 – Использование алгоритма MD5

Данный код выполняет следующие действия:

Получение входных данных из текстового поля и преобразование их в массив байтов, используя кодировку ASCII.

Создание экземпляра класса MD5CryptoServiceProvider, который предоставляет реализацию алгоритма хэширования MD5.

Измерение времени выполнения хэширования с помощью объекта «Stopwatch».

Вычисление хэш-кода для входных данных с использованием метода «ComputeHash» объекта md5. Результат сохраняется в массиве.

Итерация по каждому байту в массиве, и конвертация его в строку в шестнадцатеричном формате. Каждый шестнадцатеричный байт добавляется к текстовому полю.

Таким образом, данный код выполняет хэширование входных данных с использованием алгоритма MD5 и выводит результат хэширования и время выполнения на форму приложения.

Результат работы программы представлен на рисунке 2.2.



Рисунок 2.2 – Результат работы приложения

Также продемонстрируем что малейшее изменение, даже регистра символа меняет хеш-функцию, рисунок 2.3.

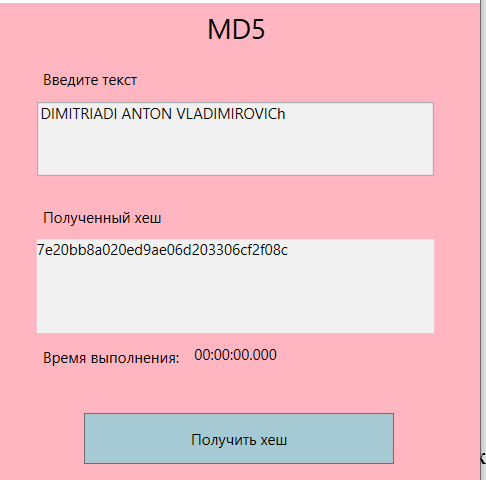


Рисунок 2.3 – Изменение хэш-функции

Далее продемонстрирован оценочный график скорости хэширования данных алгоритмом MD5, рисунок 2.4.

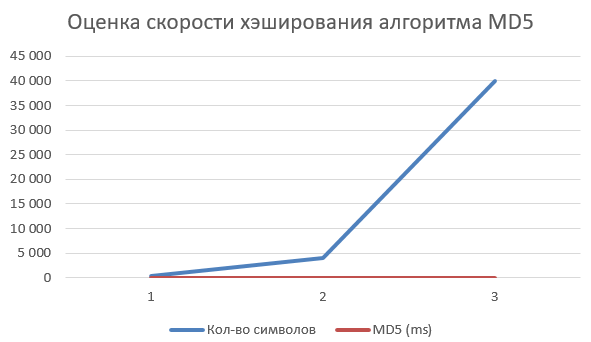


Рисунок 2.4 – График скорости хэширования алгоритма MD5

По графику можно сказать что как и полагается алгоритмам хэширования, их скорость должна быть очень быстрой, иначе алгоритм хэширования теряет свой смысл.

## Вывод

В ходе лабораторной работы были изучены алгоритмы хеширования и приобретены практические навыки их реализации и использования в криптографии. Также была оценена скорость вычисления кодов хеш-функции MD5.