Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование криптографических хеш-функций

Студент: Круглик А.В.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М.Г.

Минск 2023

## 1. Описание приложения

Приложение написано на языке программирования C# и реализиует алгоритм хеширования SHA256. Приложение обрабатывает входное сообщение, длина которого определена спецификацией

## 2. Методика выполнения поставленных задач

SHA (Алгоритмы безопасного хеширования) – это семейство криптографических хэш-функций, способных принимать сообщения произвольной длины и вычислять уникальный хэш-код фиксированной длины.

SHA-256 относится к стандарту SHA-2, использует размер слова в 32 бита. Окончание 256 означает, что фиксированный размер хэша для любого сообщения равен 256 бит.

Программная реализация алгоритма в разработанном ПС осуществляется с помощью встроенной C# библиотеки класса System.Security.Cryptography. Рассмотрим подробнее алгоритм хеширования.

Во-первых, необходимо сгенерировать так называемую «соль». «Соль»— строка данных, которая передаётся хеш-функции вместе с входным массивом данных для вычисления хэша. Программная реализация генерации соли продемонстрирована на рисунке 2.1.

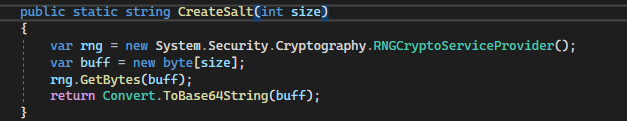


Рис. 2.1 – Генерация соли

RNG (random number generator) возвращает случайную последовательность чисел, которую мы конвертируем в строку base64. Соль используется нами для усложнения определения прообраза хэш-функции методом перебора по словарю, таким образом максимально увеличивает криптостойкость и уменьшает возможность коллизии. В данном случае использована динамическая соль, что является еще более эффективным с точки зрения профилактики атак.

Далее вызывается функция генерации хэша с помощью алгоритма SHA256, встроенная в библиотеку. Входными параметрами данной функции являются вычисленная на предыдущем шаге «соль» и входное сообщение. Программная реализация данной функции продемонстрирована на рисунке 2.2.

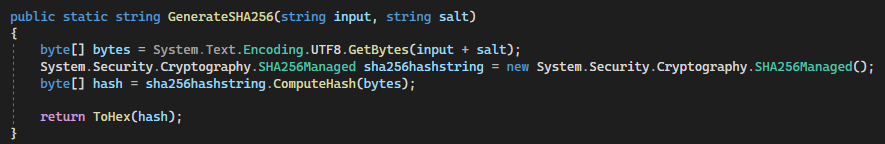


Рис. 2.2 – Реализация функции хэширования

В данном случае использована встроенная функция SHA256Managed(), позволяющая вычислить хэш. После получения результата функции, преобразуем его в 16-ричное представление с помощью разработанной нами функции ToHex() представленной на рисунке 2.3.

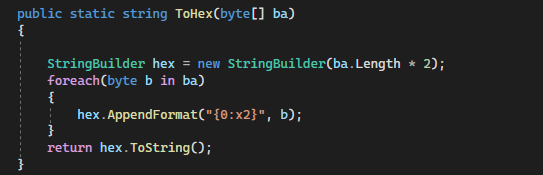


Рис. 2.3 – Реализация функции ToHex()

Рассмотрим и проанализируем результаты выполнения выше указанных функций. Результат запуска консольного приложения представлен на рисунке 2.4.

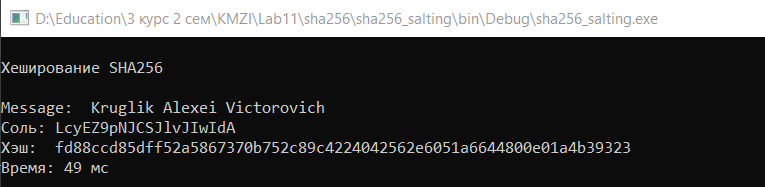


Рис. 2.4 – Результат работы приложения

Проанализировав время выполнения операции хеширования, можно предположить, что алгоритм SHA256 достаточно производительный. Это подтверждает то, что в протоколе указана максимально достиаемая скорость примерно 130 МБ/с. Ведь, если процесс вычисления хеша не достаточно быстрый, система просто не будет эффективной. График оценки времени представлен на рисунке 2.5 для текста 1000, 2000, 3000 символов.



Рис. 2.5 – Оценка времени алгоритма SHA-256

## 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 SHA256.