Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование криптографических хеш-функций**

Студент: Каспер Н.В.

ФИТ 3 курс 5 группа

Преподаватель: Берников В.О.

Минск 2020

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет:

* хешировать входное сообщение;
* оценивать скорость выполнения алгоритма.

1. **Методика выполнения поставленных задач**

MD5 – алгоритм хеширования, в котором выполняются 4 этапа по 16 раз. Каждая операция в этапе представляет собой нелинейную функцию над тремя из a, b, c и d. Затем результат добавляется к четвертой переменной, подблоку текста m и константе t. Далее результат циклически сдвигается вправо на переменное число s бит и добавляет результат к одной и переменных a, d, c и d. В конце результат заменяет одну из этих переменных. Результат хеширования – конкатенация последних значений указанных переменных – 32\*4 = 128 бит.

Результат работы приложения для алгоритма хеширования MD5 представлен на рисунке 2.1.

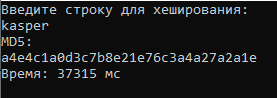


Рисунок 2.1 – Результат работы приложения для MD5

Блок кода, в котором происходит реализация алгоритма MD5 представлен на рисунке 2.2.

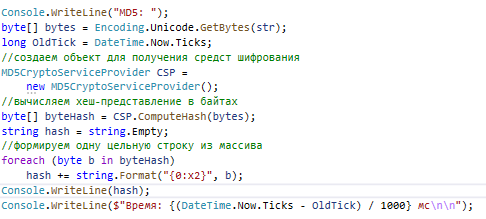


Рисунок 2.2 – Реализация MD5

SHA1 – алгоритм хеширования, в котором есть 4 этапа по 20 операций в каждом. Каждая операция представляет собой нелинейную функцию над пятью параметрами. Сдвиг и сложение делаются аналогичным образом, как в MD5. Существует алгоритм, с помощью которого блок из шестнадцати 32разрядных битных слов трансформируется в восемьдесят 32разрядных слов. Получаем 160-битную выходную последовательность, которая отображается как 16ричное число, длиной в 40 цифр.

Результат выполнения алгоритма SHA1 представлен на рисунке 2.3.



Рисунок 2.3 – Результат алгоритма SHA1

Функция для хеширования с помощью алгоритма SHA1 представлена на рисунке 2.4.

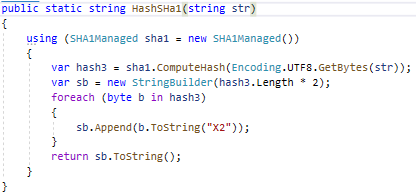


Рисунок 2.4 – Функция хеширования SHA1

SHA256 – хеш-функция из семейства алгоритмов SHA-2, которая предназначена для создания дайджестов для сообщений произвольной длины. Длина дайджеста – 256 бит. Исходное сообщение дополняется до нужной длины, а затем разбивается на блоки. Каждый блок – на 16 слов. Каждый блок сообщения пропускается через 64 итерации. На каждой итерации 2 слова преобразуются, функцию преобразования задают остальные слова. Результаты обработки каждого блока складываются, сума – значение хеш-функции. Т.к. инициализация внутреннего состояния производится результатом обработки предыдущего блока, то нет возможности обрабатывать блоки параллельно.

Результат обработки строки с помощью алгоритма SHA256 представлен на рисунке 2.5.

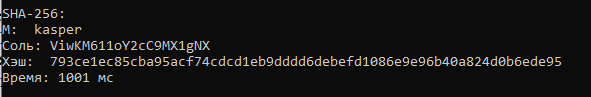


Рисунок 2.5 – Результат хеширования SHA256

Функция для хеширования строки с помощью алгоритма SHA256 представлена на рисунке 2.6.

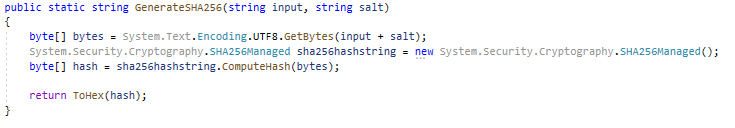


Рисунок 2.6 – Результат работы приложения для Эль-Гамаля

В алгоритмах хеширования часто используется такое понятие как соль. Соль – строка данных, которая передается хеш-функции вместе с входными данными для вычисления хеша. Используется для усложнения определения прообраза хеш-функции методом перебора по словарю возможных значений, включая радужные атаки. Позволяет скрыть факт использования одинаковых прообразов при использовании разной соли.

Таким образом, были реализованы все поставленные задачи. Были исследованы асимметричные шифры.

**Вывод**

В ходе лабораторной работы были приобретены навыки хеширования используя алгоритмы MD5, SHA1 и SHA256. Были изучены основные принципы работы хеширования.

Также было разработано приложение, на языке программирования C#, для реализации задач, связанных с хешированием данных.