МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра информационных систем и технологий

Утверждаю

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Смелов

подпись инициалы и фамилия

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

**ЗАДАНИЕ**

**к курсовому проектированию по дисциплине**"Защита информации и надежность информационных систем"

Специальность 1-40 05 01 03 «Информационные системы и технологии (издательско-полиграфический комплекс)»

Группа: 1

Студент: Шкабров Данила Сергеевич

**Тема:** Реализация и исследование алгоритма хэширования Keccak

**1. Срок сдачи студентом законченной работы**: «10» мая 2023г.

**2. Исходные данные к проекту:**

**2.1**. Функционально должны быть выполнены следующие задачи:

* Реализовать алгоритма хэширования Keccak на основе логических операций и циклических сдвигов.
* Разработать программное средство реализации данного метода или алгоритма.
* Составить руководство пользователя.

**2.2. Требования:**

* Необходимо провести аналитический обзор литературы по теме проекта
* Необходимо описать сферу применимости метода
* Программное средство может быть разработано на любом языке
* Архитектура приложения выбирается разработчиком
* Листинги проекта должны содержать комментарии

**3. Содержание расчетно-пояснительной записки:**

* Введение
* Постановка задачи
* Описание метода
* Описание программного средства
* Тестирование программного средства
* Руководство пользователя
* Заключение
* Список используемых источников
* Приложения

**4. Форма представления выполненного курсового проекта:**

* + Теоретическая часть курсового проекта должна быть представлена в формате MS Word.
  + Оформление записки должно быть согласно правилам.
  + Необходимые схемы, диаграммы и рисунки допускается делать в MS Office Visio или копии экрана (интерфейс).
  + Полные листинги проекта представляются в приложении.
  + К записке необходимо приложить CD (DVD), который должен содержать: пояснительную записку, листинги и файлы базы данных.

#### Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Введение | 24.02.2023 |  |
| 2 | Аналитический обзор литературы по теме проекта | 10.03.2023 |  |
| 3 | Разработка метода | 24.03.2023 |  |
| 4 | Разработка прототипа программного средства | 07.04.2023 |  |
| 5 | Разработка программного средства | 14.04.2023 |  |
| 6 | Тестирование программного средства | 28.04.2023 |  |
| 7 | Написание руководства пользователя | 02.05.2023 |  |
| 8 | Оформление пояснительной записки | 05.05.2023 |  |
| 9 | Сдача проекта | 10.05.2023 |  |

**5. Дата выдачи задания** «17» февраля 2023г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Д. В. Сазонова*

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата и подпись студента)

Введение

Под криптографией понимается способ передачи или хранения информации с учётом сохранения в тайне самого факта такой передачи (хранения). В отличие от криптографии, хэширование осуществляет преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины, выполняемое определённым алгоритмом. Как правило, сообщение будет выглядеть как что-либо иное.

Криптографическая хеш-функция - это математический алгоритм, который отображает данные произвольного размера в битовый массив фиксированного размера.

Результат, производимый хеш-функцией, называется «хеш-суммой» или же просто «хешем», а входные данные часто называют «сообщением».

Такой пример не часто встречается в реальной работе, но он наглядно показывает, насколько хэш-функция может облегчить работу с большими объемами информации.

Например, в массив из нескольких миллионов разных строк длиной 1 млн символов нужно добавить еще одну, при условии, что там ее еще нет. Чтобы не заниматься посимвольным сравнением каждой строки, можно предварительно вычислить хэш каждой из них, и уже сделать сравнение по нему. Вся работа упрощается и ускоряется в разы.

На основе поставленной цели были выделены следующие задачи:

* Рассмотрение хэшей разной длины;
* Проведение анализа между ними;
* Вывод результата.

1. Аналитический обзор литературы

Аналоги

MD5

MD5 — предназначен для создания контрольных сумм или «отпечатков» сообщения произвольной длины и последующей проверки их подлинности. Алгоритм MD5 основан на алгоритме MD4.

Хэш-функция MD5 генерирует 128-битное хэш-значение. Изначально она была разработана для использования в криптографии, однако со временем в ней были обнаружены уязвимости, вследствие чего для этой цели она больше не подходит. И тем не менее, она по-прежнему используется для разбиения базы данных и вычисления контрольных сумм для проверки передачи файлов.

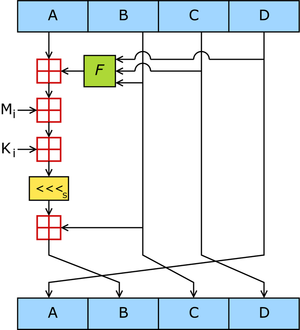


Рис. 1.1 Алгоритм работы MD5

**SHA-1**

SHA расшифровывается как Secure Hash Algorithm. SHA-1 – это первая версия алгоритма, за которой в дальнейшем последовала SHA-2.

В то время как MD5 генерирует 128-битный хэш, SHA-1 создает 160-битный (20 байт). Если представить это число в шестнадцатеричном формате, то это целое число длиной в 40 символов. Подобно MD5, этот алгоритм был разработан для криптографических приложений, но вскоре в нем также были найдены уязвимости. На сегодняшний день он считается более устойчивым к атакам в сравнении с MD5.

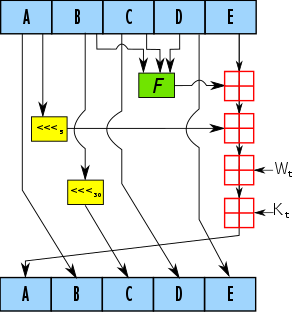


Рис. 1.2 Алгоритм работы SHA-1

**SHA-2**

Вторая версия алгоритма, SHA-2, имеет множество разновидностей. Пожалуй, наиболее часто используемая – SHA-256, которую Национальный институт стандартов и технологий (NIST) рекомендует использовать вместо MD5 и SHA-1.

Алгоритм SHA-256 возвращает 256-битное хэш-значение, что представляет собой шестнадцатеричное значение из 64 символов. Хоть это и не самый идеальный вариант, то текущие исследования показывают, что этот алгоритм значительно превосходит в безопасности MD5 и SHA-1.

Если рассматривать этот алгоритм с точки зрения производительности, то вычисление хэша с его помощью происходит на 20-30% медленнее, чем с использованием MD5 или SHA-1.

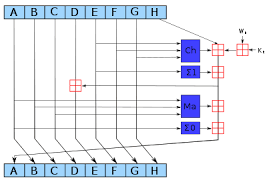


Рис. 1.3 Алгоритм работы SHA-2

2. Проектирование

В данной главе описаны технологии, которые были использованы для Реализация и исследование алгоритма хэширования Keccak. Для начала рассмотрим системы и технологии, которые мы будем использовать в данном курсовом проекте.

Windows Presentation Foundation — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML. WPF представляет собой обширный API-интерфейс для создания настольных графических программ, имеющих насыщенный дизайн и интерактивность.

C# — это объектно- и компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов.

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

1. Разработка программного средства

3.1. Создание интерфейса приложения

XAML (eXtensible Application Markup Language) - язык разметки, используемый для инициализации объектов в технологиях на платформе .NET. Применительно к WPF (а также к Silverlight) данный язык используется прежде всего для создания пользовательского интерфейса декларативным путем.

XAML – не является обязательной частью приложения, мы можем обходиться без него, создавая все элементы в файле связанного с ним кода на языке C#. Однако использование XAML все-таки несет некоторые преимущества:

* возможность отделить графический интерфейс от логики приложения, благодаря чему над разными частями приложения могут относительно автономно работать разные специалисты: над интерфейсом - дизайнеры, над кодом логики – программисты;
* компактность, понятность, простая поддержка кода.

Разработка интерфейса осуществлялась с использованием стандартных элементов, предлагаемых средой разработки Visual Studio.

* 1. Структура проекта, классы и методы

Приложение было разработано в IDE Visual Studio и представляет собой единый проект.

Самыми важными объектами в данной структуре являются объекты: MainWindow.xaml, MainWindow.xaml.cs.

MainWindow.xaml отвечает за графический интерфейс приложения. Подробно он был описан в предыдущем разделе.

MainWindow.xaml.cs является классом C#, который инициализирует программный компонент, а также содержит обработчики событий для всех UI элементов интерфейса.

Именно этот класс с помощью описанных методов может определить размер, генерировать и отображать хэш.