Министерство образования Республики Беларусь

Учреждение образования БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы защиты информации

Отчёт по лабораторной работе №5

По теме «Хэш-функции»

Выполнили: студент гр. 053501 Кривецкий Р. А.

Проверил: ассистент кафедры информатики Лещенко Е. А.

Минск 2023

**СОДЕРЖАНИЕ**

[1 Теоретические сведения 3](#_Toc148061125)

[2 Программная реализация 4](#_Toc148061126)

[Заключение 5](#_Toc148061127)

[Приложение А(обязательное) Листинг программного кода 6](#_Toc148061128)

# 1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

«Стрибог» (англ. STREEBOG) — криптографический алгоритм вычисления хеш-функции с размером блока входных данных 512 бит и размером хеш-кода 256 или 512 бит.

Описывается в ГОСТ 34.11-2018 «Информационная технология. Криптографическая защита информации. Функция хэширования» — действующем межгосударственном криптографическом стандарте.

Разработан Центром защиты информации и специальной связи ФСБ России с участием ОАО «ИнфоТеКС» на основе национального стандарта Российской Федерации ГОСТ Р 34.11-2012 и введен в действие с 1 июня 2019 года приказом Росстандарта № 1060-ст от 4 декабря 2018 года.

Стандарт ГОСТ Р 34.11-2012 разработан и введён в качестве замены устаревшему стандарту ГОСТ Р 34.11-94:

Название хеш-функции — «Стрибог», по имени славянского божества, — часто используется вместо официального названия стандарта, хотя в его тексте явно не упоминается.

Алгоритм хэширования GOST 34.11-2018 (также известный как Streebog) является российским стандартом для создания хеш-функций. Он предоставляет надежный метод преобразования произвольного входного сообщения в фиксированный хеш-значение определенной длины. Принцип работы этого алгоритма можно описать следующим образом:

1. Исходное сообщение: На вход алгоритму подается произвольное сообщение в виде битовой последовательности. GOST 34.11-2018 поддерживает сообщения длиной от 1 до 2^64-1 битов.

2. Инициализация: Для начала хэширования, исходное сообщение разбивается на блоки фиксированной длины. Размер блока зависит от выбранной длины хеш-значения, которая может быть 256 бит или 512 бит. Все хеш-значения задаются как кратные 8 битам.

3. Инициализационный вектор: Для начала вычислений используется некоторое начальное значение, известное как инициализационный вектор. Инициализационный вектор устанавливается в соответствии с выбранной длиной хеш-значения.

4. Обработка блоков: Каждый блок сообщения обрабатывается независимо друг от друга. В процессе обработки блока выполняются несколько раундов преобразований, которые включают в себя подстановки, перестановки и операции над битами. Эти раунды выполняются для каждого блока и включают в себя взаимодействие с инициализационным вектором и предыдущими обработанными блоками.

5. Финальная обработка: После того как все блоки обработаны, результаты объединяются в конечное хеш-значение. Это значение можно рассматривать как фиксированное представление всего входного сообщения.

6. Выдача хеш-значения: Полученное хеш-значение является окончательным результатом работы алгоритма и может использоваться для проверки целостности данных, аутентификации и других приложений, где требуется надежное и уникальное представление исходной информации.

MD5 (англ. Message Digest 5) — 128-битный алгоритм хеширования, разработанный профессором Рональдом Л. Ривестом из Массачусетского технологического института (Massachusetts Institute of Technology, MIT) в 1991 году. Предназначен для создания «отпечатков» или дайджестов сообщения произвольной длины и последующей проверки их подлинности. Широко применялся для проверки целостности информации и хранения хешей паролей.

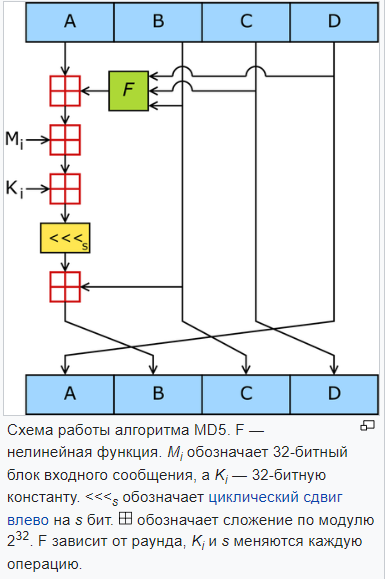


Рисунок 1.1 — Схема работы алгоритма

Схема работы алгоритма МД5 показана на рисунке 1.1.

На вход алгоритма поступает входной поток данных, хеш которого необходимо найти. Длина сообщения измеряется в битах и может быть любой (в том числе нулевой). Запишем длину сообщения в L. Это число целое и неотрицательное. Кратность каким-либо числам необязательна. После поступления данных идёт процесс подготовки потока к вычислениям.

Ниже на рисунках 1.2-1.6 показаны 5 шагов алгоритма.

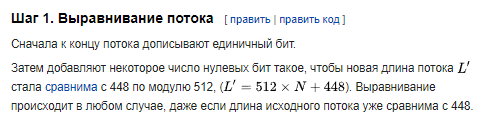


Рисунок 1.2 — Шаг 1

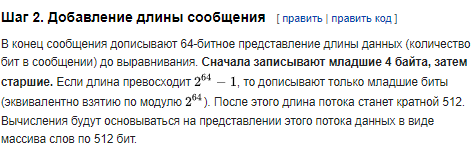


Рисунок 1.3 — Шаг 2

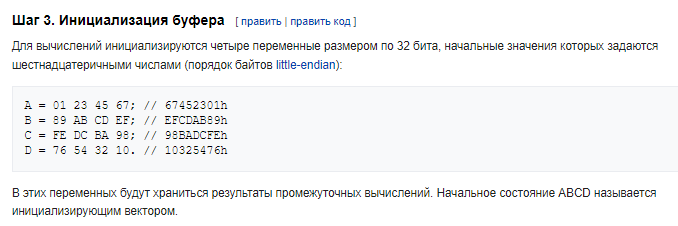
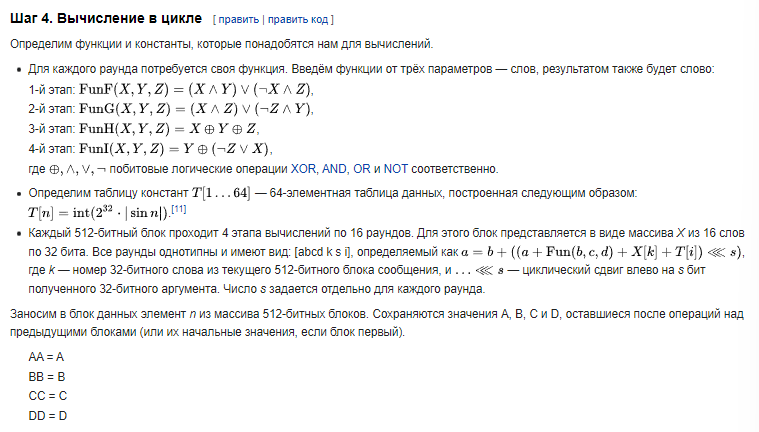


Рисунок 1.4 — Шаг 3



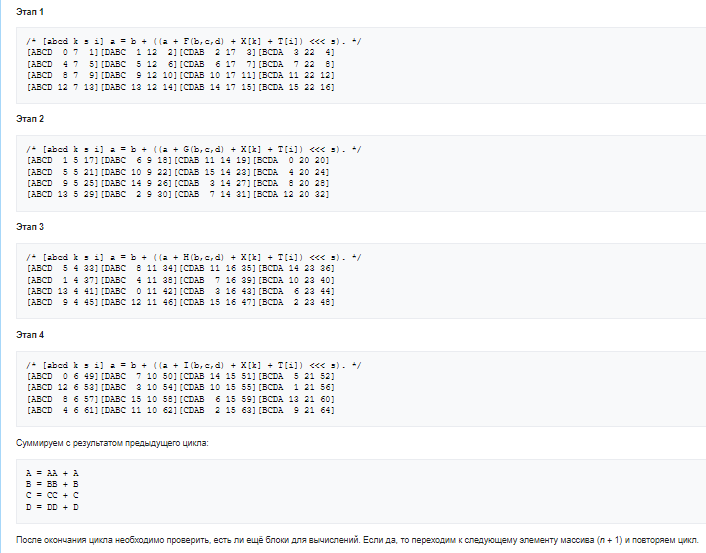


Рисунок 1.5 — Шаг 4

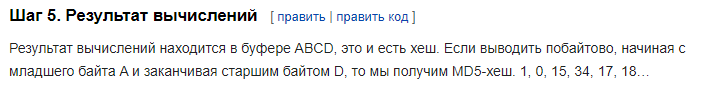


Рисунок 1.6 — Шаг 5

**2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

Необходимо реализовать программное средство контроля целостности сообщений с помощью вычисления хэш-функции и алгоритма ГОСТ 34.11. Необходимо выполнить 2 реализации хэш-функций: ГОСТ 34.11 и MD5.

Программное средство реализовано на языке программирования Python. Демонстрация работы показана на рисунке 2.1



Рисунок 2.1 — Демонстрация работы

Вывод в консоль сделан для дополнительной наглядности, показано на рисунке 2.2.

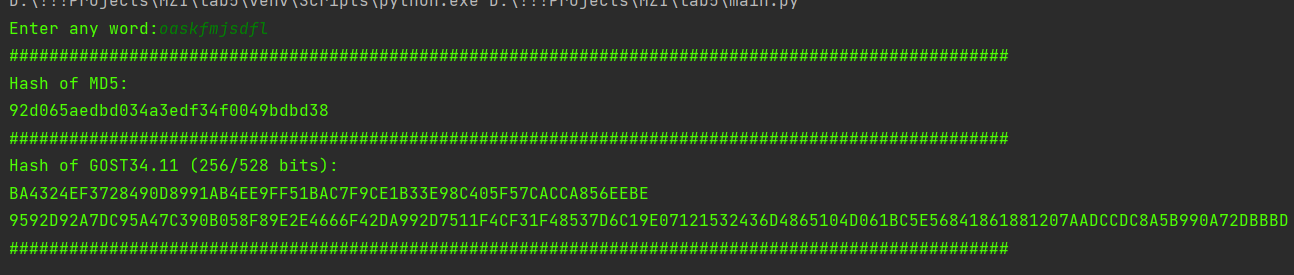


Рисунок 2.2 — Вывод хэш-функций

# ЗАКЛЮЧЕНИЕ

GOST 34.11-2018 обеспечивает высокую степень стойкости к различным атакам, таким как коллизии и предобразования. Алгоритм также прошел серьезное криптографическое исследование и анализ, чтобы удостовериться в его надежности.

MD5 (Message Digest Algorithm 5) был широко использован для хеширования и проверки целостности данных, но с течением времени стал уязвим к коллизиям и предобразованиям. В настоящее время, MD5 не рекомендуется использовать для криптографических целей, так как существуют более безопасные альтернативы, такие как SHA-2 и SHA-3. MD5 считается устаревшим и уязвимым к различным атакам, и его использование в критических системах не рекомендуется.

# ПРИЛОЖЕНИЕ А

(**обязательное)**

**Листинг программного кода**

Файл main.py

from GOST3411.GOST3411 import GOST3411

from MD5.MD5 import MD5

def main():

word = input("Enter any word:")

md5\_hash = MD5.hash(word)

print("#" \* 100)

print("Hash of MD5:")

print(md5\_hash)

print("#" \* 100)

print("Hash of GOST34.11 (256/528 bits):")

print(GOST3411().hash(word, 256))

print(GOST3411().hash(word, 512))

print("#" \* 100)

if \_\_name\_\_ == "\_\_main\_\_":

main()