МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КУБГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчет**

**по практическому заданию №11**

**по курсу**

**«КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ»**

Работу выполнил

Студенты 46 группы

Прозоров М.С.

Преподаватель:

Крамаренко А.А.

Краснодар 2024

**Постановка задачи**

Реализовать программный продукт построения sha-384 для введенного текста.

SHA-384 является одним из алгоритмов из семейства Secure Hash Algorithms (SHA), разработанных Национальным институтом стандартов и технологий (NIST) США. SHA-384, вместе с SHA-512, SHA-224 и SHA-256, является частью SHA-2 семейства хэш-функций. SHA-384 генерирует хэш-значение размером 384 бита (или 48 байт) из входных данных произвольной длины. Вот как работает алгоритм SHA-384:

Инициализация:

Алгоритм начинается с инициализации внутренних переменных (иногда называемых "регистрами"). В SHA-384 для этого используются 8 инициализационных векторов, специфичных для SHA-384, каждый из которых является 64-битным числом.

Предварительная обработка:

Добавление бита 1 к сообщению: cначала к сообщению добавляется бит 1.

Добавление битов 0: затем добавляется некоторое количество битов 0, чтобы длина сообщения (в битах) стала на 128 бит меньше следующего кратного 1024 битам.

Добавление длины сообщения: В конец сообщения добавляется 128-битное представление исходной длины сообщения (до добавления битов), так что общая длина сообщения теперь кратна 1024 битам.

Основной цикл

SHA-384 использует итеративную структуру, где входное сообщение делится на блоки по 1024 бита (128 байт). Для каждого блока выполняются следующие шаги:

Расширение блока: Каждый блок из 1024 бит расширяется до 80 64-битных слов при помощи специальных операций. Это делается для увеличения эффективности и безопасности хэширования.

Основной цикл: для каждого из 80 раундов используется комбинация битовых сдвигов, битовых вращений и логических операций с текущими значениями внутренних переменных и расширенными словами. В каждом раунде используется также константа, уникальная для каждого раунда.

Обновление состояния: после выполнения всех 80 раундов значения внутренних переменных обновляются.

Финализация

После обработки всех блоков исходного сообщения конечное состояние внутренних переменных представляет собой 384-битный хэш. Это делается путём конкатенации первых 6 из 8 итоговых 64-битных слов (переменных состояния).

**Вывод**

Реализовал программный продукт построения sha-384 для введенного текста.

**Листинг:**

import hashlib

def generate\_sha384\_hash(input\_text):

# Объект хеша SHA-384

sha384\_hash = hashlib.sha384()

# Кодирование введённого текста в байты и обновление объекта хэша

sha384\_hash.update(input\_text.encode('utf-8'))

# Получение хэша в виде шестнадцатеричной строки

hex\_dig = sha384\_hash.hexdigest()

return hex\_dig

input\_text = input("Введите текст для генерации SHA-384 хэша: ")

print(f'Введенный текст: {input\_text}')

print("SHA-384 хэш введённого текста:", generate\_sha384\_hash(input\_text))