Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Информационная безопасность**

Студент: Козека Е. М.

ФИТ 3 курс 4 группа

Преподаватель: Нистюк О. А.

Минск 2025

**Лабораторная работа №11. Исследование криптографических хеш-функций**

**Цель:** исследование алгоритмов хеширования и приобретение практических навыков их реализации и использования в криптографии.

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию и алгоритмам реализации операций вычисления однонаправленных хеш-функций.
2. Освоить методику оценки криптостойкости хеш-преобразований на основе «парадокса дня рождения».
3. Разработать приложение для реализации заданного алгоритма хеширования (из семейств MD и SHA).
4. Оценить скорость вычисления кодов хеш-функций.
5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Теоретические сведения**

*Хеш-функция* – математическая или иная функция *h* = *H*(*М*), которая принимает на входе строку символов *М*, называемую также *прообразом*, переменной длины *n* и преобразует ее в выходную строку фиксированной (обычно – меньшей) длины *l*.

*Хеширование* – это преобразование входного массива данных определенного типа и произвольной длины (практически) в выходную битовую строку фиксированной длины.

Преобразования называются *хеш-функциями,* или *функциями свертки*, а их результаты называют *хешем*, *хеш-кодом*, *хеш-таблицей* или *дайджестом сообщения*.

Все существующие функции хеширования можно разделить на два больших класса:

* бесключевые хеш-функции, зависящие только от сообщения;
* хеш-функции с секретным ключом, зависящие как от сообщения, так и от секретного ключа.

*Криптографическая хеш-функция* – это специальный класс хеш-функций, который имеет различные свойства, необходимые для решения задач в области криптографии.

*Основные задачи, решаемые с помощью хеш-функций***:**

1. аутентификация (хранение паролей);
2. проверка целостности данных;
3. защита файлов;
4. обнаружение зловредного ПО;
5. криптовалютные технологии.

К основным свойствам хеш-функций можно отнести следующие.

*Свойство 1*. *Детерминированность*: независимо от того, сколько раз вычисляется *H*(*M*), *M* – const, при использовании одинакового алгоритма код хеш-преобразования *h* всегда должен быть одинаковым.

*Свойство 2*. *Скорость вычисления* хеша *h*: если процесс вычисления *h* недостаточно быстрый, система просто не будет эффективной.

*Свойство 3*. *Сложность обратного вычисления*: для известного *H*(*М*) невозможно (практически) определить *М*. Это важнейшее свойство хеш-функции для криптографических применений – *свойство односторонности преобразования*.

*Свойство 4*. Даже минимальные изменения в хешируемых данных (*М ≠ М'*) должны изменять хеш: *Н*(*M*) ≠ *Н*(*М'*).

*Свойство* 5. *Коллизионная устойчивость* (стойкость).

*Коллизией хеш-функции Н* называют ситуацию, при которой различным входам (в общем случае – *х* и *у* или *М* ≠ *М'*) соответствует одинаковый хеш-код: *H*(*x*) = *H*(*y*) или *H*(*М*) = = *H*(*М'*).

Зная *М*, трудно найти такое *М'* (*М* ≠ *М'*), для которого *H*(*М*) = *H*(*М'*). Если последнее равенство выполняется, то говорят о *коллизии 1-го рода*. Если *случайным образом* выбраны два сообщения (*М* и *М'*), для которых *H*(*М*) = *H*(*М'*), говорят о *коллизии 2-го рода*.

*Мерой криптостойкости хеш-функции считается вычислительная сложность нахождения коллизии.*

Для хеш-функций одним из основных средств поиска коллизий является метод, основанный на известной статистической задаче – «парадоксе дня рождения».

Основной постулат *парадокса «дней рождения»* гласит: в группе минимум из 23 человек с вероятностью более 0,5 день рождения одинаков хотя бы у двух членов группы. Парадоксом является высокая (как кажется на первый взгляд) вероятность наступления указанного события. При этом предполагается, что:

• в этой группе нет близнецов;

• люди рождаются независимо друг от друга, т. е. дата (день) рождения любого человека не влияет на дату рождения другого;

• люди рождаются равномерно и случайно, т. е. люди с равной вероятностью могут рождаться в любой день года; с формальной точки зрения это означает, что вероятность *р*1 рождения отдельно выбранного члена группы (как и любого человека) в любой выбранный день равна *р*1 *=* 1 / 365 (хотя известно, что в реальности рождение людей не совсем соответствует такому предположению).

Хеш-функция – это функция, выполняющая отображение из множества *М*в число, находящееся в интервале [0, *m* – 1]: *h*: *M*→ [0, *m* – 1].

Алгоритмы семейства MD-x (2/4/5/6) являются творениями Р. Ривеста; MD – Message Digest. Алгоритм MD6, в отличие от предыдущих версий алгоритма этого семейства, не стандартизован.

Алгоритмы семейства SHA (SHA – Secure Hash Algorithm) являются в настоящее время широко распространенными. По существу, во многих случаях завершился переход от SHA-1 к стандартам версии SHA-2. SHA-2 – собирательное название алгоритмов SHA- 224, SHA-256, SHA-384 и SHA-512. SHA-224 и SHA-384 являются, по сути, аналогами SHA-256 и SHA-512 соответственно.

**Ход работы**

**Задание 1.** Разработать оконное приложение, реализующее один из алгоритмов хеширования из указанного преподавателем семейства (MD или SHA; или иного).

Функция хеширования алгоритмом SHA-256 принимает на вход оригинальный текст и осуществляет хеширование.

app.post('/sha256', (req, res) => {

    const message = req.body.message;

    if (!message) {

        return res.status(400).json({ error: 'Message is required' });

    }

    let startTime = performance.now();

    const hash =

crypto.createHash('sha256').update(message).digest('hex');

    let endTime = performance.now();

    const hashingTime = endTime - startTime;

    const length = hash.length / 2;

    res.json({ hash, length, hashingTime });

});

Листинг 11.1 – Функция хеширования алгоритмом SHA-256

Результат работы функции с исходным сообщением «Kozeka Elizaveta Maksimovna» представлен на рисунке 11.1.

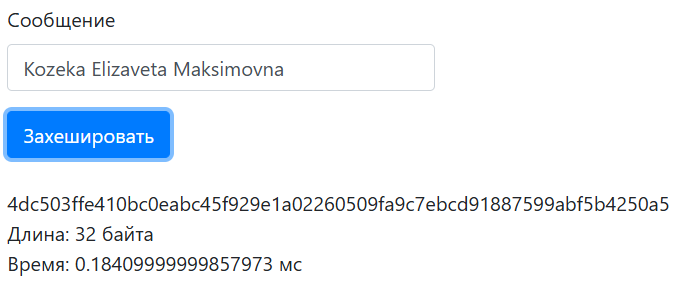


Рисунок 11.1 – Результат работы функции хеширования SHA-256

Длина хеша – 32 байта или 256 бит. Время хеширования сообщения – менее 1 мс, что отражает высокую временную эффективность алгоритма.

Функция хеширования алгоритмом MD-5 принимает на вход оригинальный текст и осуществляет хеширование.

app.post('/md5', (req, res) => {

    const message = req.body.message;

    if (!message) {

        return res.status(400).json({ error: 'Message is required' });

    }

    let startTime = performance.now();

    const hash =

crypto.createHash('md5').update(message).digest('hex');

    let endTime = performance.now();

    const hashingTime = endTime - startTime;

    const length = hash.length / 2;

    res.json({ hash, length, hashingTime });

});

Листинг 11.2 – Функция хеширования алгоритмом MD-5

Результат работы функции с исходным сообщением «Kozeka Elizaveta Maksimovna» приведен на рисунке 11.2.

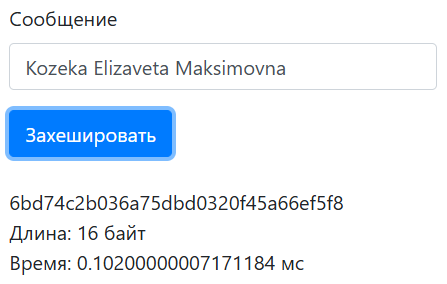


Рисунок 11.2 – Результат работы функции хеширования MD-5

Длина хеша – 16 байт или 128 бит. Время зашифрования сообщения – менее 1 мс, что сравнимо с алгоритмом SHA-256.

**Вывод:** в ходе выполнения лабораторной работы был изучен принцип реализации алгоритмов хеширования SHA-256 и MD5. Также было разработано приложение, выполняющее хеширования на основе данных алгоритмов. Была оценена скорость хеширования различными алгоритмами.