СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ ХЕШИРОВАНИЯ

Ковалевич М.А., Филипченко К.В., студенты гр.361401

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Руспублика Беларусь

Тимофеев А.М. – канд. техн. наук, доцент

Аннотация. Произведен сравнительный анализ алоритмов хеширования ГОСТ 34.11-2018, СТБ 34.101.77-2016, SHA-256 и BLAKE2. С учетом критериев сравнения была сформирована оценка уровня информационной безопасности приведенных алгоритмов. На основе анализа был разработан программный модуль HashComparator, который сравнивает скорость вычисления криптографических хешей SHA-256 и BLAKE2b для любого заданного файла и рекомендует более быстрый алгоритм. Код решает задачу объективного выбора оптимального алгоритма хеширования для конкретного файла в зависимости от времени выполнения – важного критерия в приложениях, работающих с большими объёмами данных.

Ключевые слова. сравнительный анализ, оценка уровня информационной безопасности, алгоритм хеширования, ГОСТ 34.11-2018, СТБ 34.101.77-2016, SHA-256, BLAKE2.

Криптографические хеш-функции – это математический алгоритм, преобразовывающий произвольный массив данных в состоящую из букв и цифр строку фиксированной длины.

ГОСТ 34.11-2018 – это межгосударственный стандарт, который описывает криптографическую функцию хеширования «Стрибог» (Streebog), является обязательным для использования в системах криптографической защиты информации на территории России. Алгоритм также известен как «Стрибог» и поддерживает два уровня безопасности – 256 бит и 512 бит фиксированной длины. Основная сфера применения включает обеспечение целостности данных, проверку подлинности информации, а также является неотъемлемой частью процедур формирования и проверки электронной цифровой подписи (ЭЦП) по ГОСТ 34.10-2018. Документ действует с 1 июня 2019 года, а его разработчиками выступили Центр ФСБ России и компания «ИнфоТеКС» [1].

В Республике Беларусь государственным стандартом, который описывает алгоритмы хеширования, стандартизированные в Республике Беларусь, является Bash. Полное название стандарта – СТБ 34.101.77-2016 «Информационные технологии и безопасность. Алгоритмы хеширования». Настоящий стандарт определяет семейство криптографических алгоритмов хеширования, предназначенных для контроля целостности и необратимого сжатия данных. Алгоритмы хеширования настоящего стандарта отличаются уровнем стойкости l. Это натуральное число, кратное 16 и не превосходящее 256. Алгоритм уровня l вычисляет хеш-значения длины $2l$, обрабатывая входные слова блоками длины $1536 - 4l$. Уровни l = 128, l = 192 и l = 256 являются стандартными, им следует отдавать предпочтение [2].

Алгоритм хеширования BLAKE2 предназначен для обеспечения высокой скорости хеширования при сохранении высокого уровня безопасности, и он может быть использован в различных приложениях, таких как аутентификация, проверка целостности данных и многие другие криптографические задачи [3].

Алгоритм хеширования SHA-256, входящий в семейство SHA-2, является одним из самых популярных и надежных алгоритмов. Он генерирует фиксированный 256-битный хэш, что делает его устойчивым к атакам на коллизии и подбор. SHA-256 широко используется в блокчейн-системах, сертификатах SSL/TLS, а также для проверки целостности файлов [4].

Сравнительный анализ перечисленных алгоритмов осуществлялся по следующим критериям: скорость вычисления хеша, коллизионная стойкость, лавинный эффект, длина хеша, криптостойкость, равномерность распределения, вычисление алгоритма. Выбранные критерии для сравнения хеш-алгоритмов являются общепринятыми в криптографическом анализе и системной инженерии. Скорость вычисления определяет применимость алгоритма в реальных системах с ограниченными ресурсами. Коллизионная стойкость представляет собой фундаментальное требование, вытекающее из определения криптографической хеш-функции, и при её нарушении алгоритм считается скомпрометированным. Лавинный эффект характеризует качество перемешивания входных данных и необходим для предотвращения статистического и дифференциального криптоанализа. Длина хеша напрямую влияет на стойкость к атакам полного перебора и парадокса дней рождений, при этом современные стандарты предписывают минимальную длину 256 бит. Криптостойкость включает устойчивость к атакам удлинения сообщения, что критически важно для использования в HMAC. Равномерность распределения обеспечивает статистическую независимость выхода от входа. Способ вычисления выбран для анализа архитектурных особенностей, влияющих на аппаратную оптимизацию и устойчивость к различным классам атак. Результат сравнительного анализа представлен в таблице 1.

Таблица 1 – Обзор алгоритмов хеширования

Критерии для сравнения	ГОСТ 34.11-2018	СТБ 34.101.77-2016	SHA-256	BLAKE2
Скорость вычисления	100 МБ/с для 1 потока	150 МБ/с для 1 потока	200-400 МБ/с для 1 потока	200-400 МБ/с для 1 потока
Коллизионная стойкость	$2^{n/2}$, n-длина хэша	$2^{n/2}$, n-длина хэша	$2^{n/2}$, n-длина хэша	$2^{n/2}$, n-длина хэша
Лавинный эффект	S-блоки, P-преобразование, L-преобразование, 12 раундов	Sponge-конструкция и раундовая функция bash-f	Основа на ARX, функции перемешивания, 64 раунда	Основа на ARX функция перемешивания G, 10/12 раундов
Длина хеша	512 бит, 256 бит	512 бит, 384 бита, 256 бит	256 бит	512 бит, 256 бит
Криптостойкость	Устойчивость к Length Extension (спец.конструкция)	Устойчивость к Length Extension	–	Устойчивость к Length Extension (изначально заложено)
Равномерность распределения	Равномерность обеспечивается SPN-структурой	Sponge-конструкция с перестановкой bash-f (24 раунда)	Равномерность обеспечивается ARX-архитектурой	Равномерность обеспечивается ARX-архитектурой
Вычисление	Разбиение на блоки по 512 бит, 12 раундов преобразований, финальное преобразование	Sponge-конструкция: абсорбция блоков сообщения в 1536-битное состояние через XOR с последующей 24-раундовой перестановкой bash-f, затем сквиз хэша длиной 2l бит (256/384/512).	Разбиение на блоки, создание расписания ключей, 64 раунда сжатия	Инициализация состояния (матрица 4x4), функция сжатия G, раунды перемешивания

На основе представленных данных все четыре алгоритма обеспечивают приемлемый уровень информационной безопасности, однако между ними есть существенные различия. По коллизионной стойкости все алгоритмы соответствуют теоретическому пределу $2^{n/2}$ и не имеют явных уязвимостей. Ключевым дифференцирующим фактором выступает устойчивость к атакам удлинения сообщения: ГОСТ 34.11-2018, СТБ 34.101.77-2016 и BLAKE2 устойчивы к данному классу атак, тогда как SHA-256 уязвим, что снижает его безопасность при использовании в протоколах типа HMAC. По лавинному эффекту SHA-256 с 64 раундами обеспечивает наибольший консервативный запас прочности, проверенный временем, тогда как ГОСТ с 12 раундами и BLAKE2 с 10-12 раундами могут иметь меньший запас, хотя их архитектуры (SPN и ARX соответственно) считаются стойкими. По длине хеша SHA-256 предлагает только 256 бит, тогда как остальные алгоритмы поддерживают 512 бит, что обеспечивает значительно более высокую стойкость к атакам на квантовых компьютерах.

Выбор SHA-256 и BLAKE2 для разработанной программы обоснован их широкой доступностью для конечных пользователей и отсутствием необходимости в дополнительных криптографических библиотеках. Оба алгоритма входят в стандартную библиотеку hashlib языка Python, что обеспечивает кроссплатформенную совместимость и простоту развертывания без установки стороннего программного обеспечения. SHA-256 присутствует во всех современных операционных системах и языках программирования. BLAKE2 включен в стандартную библиотеку Python начиная с версии 3.6, что делает его доступным. На рисунке 1 представлена блок-схема разработанного программного модуля.

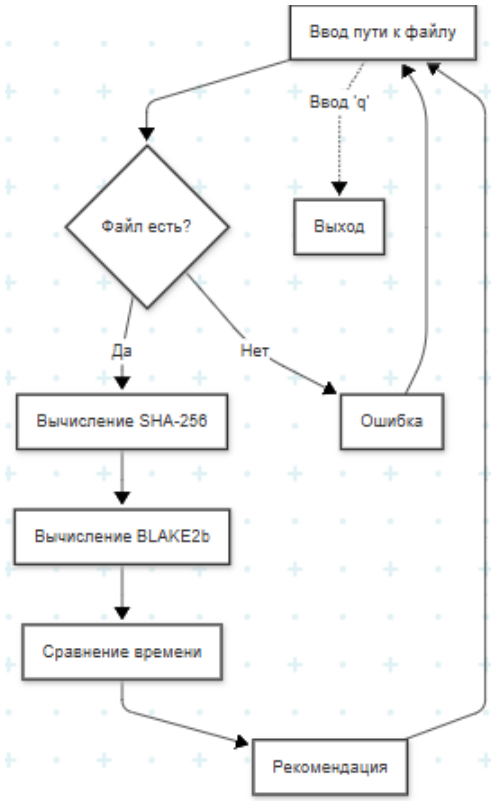

Рисунок 1 – Блок-схема программного модуля HashComparator

Разработанная программа представляет собой инструмент для эмпирического сравнения производительности двух криптографических хеш-функций (SHA-256 и BLAKE2b) на произвольном файле, выбранном пользователем. Листинг модуля включает определение класса HashComparator, который инкапсулирует логику вычисления хешей, замера времени и формирования рекомендации. При инициализации объекта сохраняется путь к целевому файлу и инициализируется пустой словарь results для накопления результатов измерений. Методы calculate_hash_sha256 и calculate_hash_blake реализуют измерение времени выполнения соответствующих алгоритмов с использованием высокоточного таймера time.perf_counter, что позволяет минимизировать погрешности, связанные с системными вызовами. Чтение файла осуществляется блоками по 4096 байт, что обеспечивает баланс между скоростью ввода-вывода и потреблением оперативной памяти. Вычисленный хеш возвращается в шестнадцатеричном формате вместе с затраченным временем. Метод run_analysis последовательно вызывает оба измерительных метода, выводя на экран информацию об анализируемом файле (его имя и размер в байтах), а также промежуточные результаты для каждого алгоритма. После завершения замеров вызывается метод get_recommendation, который находит алгоритм с минимальным временем выполнения, выводит его название и значение времени, а также формирует отсортированную по скорости таблицу сравнения.

Таким образом, наивысший уровень безопасности демонстрируют ГОСТ 34.11-2018, BLAKE2 и СТБ 34.101.77-2016 как современные стандартизированные алгоритмы с поддержкой 512-битного хеша и устойчивостью к удлинению сообщения, тогда как SHA-256, несмотря на широкое распространение, уступает по ряду криптографических свойств. Выбор SHA-256 и BLAKE2 для программного модуля обусловлен их наличием в стандартной библиотеке Python, что обеспечивает доступность для пользователей без установки дополнительного ПО.

Список использованных источников:

1. Электронный фонд правовых и нормативно-технических документов : [сайт]. – Мск., 2019-2026. – URL: https://docs.cntd.ru/document/1200161707 (дата обращения: 03.04.2026).

2. Википедия : [сайт]. – Мн., 2016-2026. – URL: https://ru.wikipedia.org/wiki/Bash_(хэш-функция) (дата обращения: 03.04.2026).

3. Server Flow : [сайт]. – Мск., 2025-2026. – URL: https://serverflow.ru/blog/stati/sovremennye-algoritmy-kheshirovaniya-ot-sha-256-do-argon2/ (дата обращения: 03.04.2026).

4. Хабр : [сайт]. – Мск., 2023-2026. – URL: https://habr.com/ru/articles/729260/ (дата обращения: 03.04.2026).