Инженерный вестник Дона, №8 (2025)
ivdon.ru/ru/magazine/archive/n8y2025/10303

Эффективность применения некоторых алгоритмов хеширования для веб-приложения с интерфейсами регистрации и аутентификации пользователей

А.Р. Айдинян, О.Л. Легонько, Н.С. Мамонов

Донской государственный технический университет, Ростов-на-Дону

Аннотация: В статье приведены результаты исследования эффективности алгоритмов хеширования Argon2, Scrypt и Bcrypt в контексте разработки веб-приложений с функциями регистрации и аутентификации пользователей. Основное внимание в данном исследовании уделено анализу устойчивости алгоритмов к атакам методом перебора, аппаратным атакам (GPU/ASIC), а также оценке их вычислительной производительности. Результаты экспериментов демонстрируют преимущества Scrypt в контексте баланса между временем выполнения и безопасностью. Предложены рекомендации по выбору алгоритмов в зависимости от требований к безопасности и производительности.
Ключевые слова: алгоритм хеширования, интерфейс регистрации пользователя, интерфейс аутентификации пользователя, защита конфиденциальных данных.

Введение

Алгоритмы хеширования являются критически важным компонентом систем аутентификации, обеспечивающим защиту конфиденциальных данных пользователей. Современные криптографические стандарты требуют применения устойчивых к брутфорсу и аппаратным атакам методов, таких как Argon2, Scrypt и Bcrypt. Несмотря на активное использование этих алгоритмов, их сравнительная эффективность в условиях реального веб-приложения остаётся недостаточно изученной. Целью данного исследования является оценка производительности и безопасности указанных алгоритмов при интеграции в систему регистрации |и аутентификации, разработанную на базе фреймворка Flask. Актуальность работы обусловлена необходимостью оптимизации выбора алгоритмов хеширования для минимизации рисков утечки данных.

Алгоритм хеширования паролей представляет собой математический процесс, в ходе которого пароль в виде обычного текста преобразуется в неразборчивую строку символов. Эта строка, или хеш, хранится в базе

данных вместо обычного пароля. Этот процесс известен как одностороннее шифрование, так как оригинальный пароль не может быть получен из хеша, что делает невозможным доступ хакеров к учётной записи пользователя [1].

Обзор алгоритмов хеширования

Современные исследования в области криптографии [1–3] подтверждают важность адаптивных алгоритмов хеширования, способных противостоять эволюции вычислительных мощностей. Argon2, победитель конкурса Password Hashing Competition 2015, позиционируется как алгоритм с максимальной устойчивостью к GPU- и ASIC-атакам за счёт гибкой настройки параметров памяти и параллелизма [4]. Scrypt, разработанный для защиты от аппаратных атак [5], использует интенсивное потребление памяти, что увеличивает сложность брутфорса. Bcrypt, основанный на алгоритме Blowfish, обеспечивает защиту за счёт адаптивной сложности вычислений [6], однако уступает в устойчивости к современным атакам. В работах [7–8] отмечается, что комбинация хеширования с дополнительными мерами безопасности (например, двухфакторной аутентификацией) существенно снижает риски компрометации данных.

Применение алгоритмов хеширования

Авторами было разработано веб-приложение с интерфейсами регистрации и аутентификации пользователей с применением алгоритмов хеширования Argon2, Scrypt и Bcrypt.

Целью разработки веб-приложения было сравнение эффективности этих алгоритмов, а также попытка реализовать систему регистрации и аутентификации пользователей с использованием этих алгоритмов. При этом было принято во внимание следующее.

Argon2 – разработан для максимальной устойчивости к атакам методом подбора и аппаратным атакам.

К его преимуществам можно отнести наличие:

– широких возможностей настройки (памяти, времени, параллелизма);

– трех вариантов, а именно Argon2d, Argon2i, Argon2id;

– комплексной защиты от атак на GPU, ASIC.

Scrypt – более современный алгоритм, который использует память для увеличения стоимости вычислений. Это делает его более устойчивым к атакам с использованием специализированного оборудования, такого как ASIC [7].

К преимуществам этого алгоритма можно отнести:

– наличие настраиваемого ЦП и расход памяти;

– обеспечение надежной защиты от атак ASIC и GPU;

– использование в некоторых криптовалютах (например, Litecoin).

Bcrypt – один из наиболее известных алгоритмов для хеширования паролей. Он использует соль (соль – это случайная строка данных, которая добавляется к паролю перед хешированием) и повторные циклы хеширования, что делает его устойчивым к атакам с использованием радужных таблиц.

Преимущество Bcrypt обеспечивается использованием:

– коэффициента адаптивной работы, который может увеличить вычислительные затраты по мере совершенствования оборудования;

– встроенной соли для устойчивости к радужному эффекту;

– широкой поддержки и простоты внедрения.

В таблице 1 представлены основные свойства алгоритмов хеширования Argon2, Scrypt и Bcrypt.

Таблица № 1

Основные свойства алгоритмов хеширования Argon2, Scrypt и Bcrypt

Алгоритм	Основные моменты безопасности	Типичное время хеширования	Использование памяти
Argon2	Самый безопасный, устойчивый к памяти, устойчивый к GPU/ASIC	~150 мс	~64 MB
Scrypt	Очень безопасный, устойчивый к памяти, устойчивый к ASIC	~200 мс	~32 MB
Bcrypt	Безопасный, но с фиксированной памятью, менее устойчивый к GPU	~250 мс	~4 KB

В разработанном авторами веб-приложении предусмотрены:

– регистрация пользователей, то есть пользователи могут зарегистрироваться, указав имя пользователя и пароль. При этом для каждого пароля генерируется хеш с использованием всех трех алгоритмов, а соль для хеширования генерируется случайным образом;

– аутентификация пользователей, когда всякий раз при попытке входа в систему проверяется, совпадает ли введенный пароль с хешем, хранящимся в базе данных для каждого алгоритма хеширования;

– сравнение времени работы алгоритмов, а именно реализована функция для измерения времени работы каждого алгоритма хеширования, что позволяет выбрать наиболее быстрый алгоритм для использования в приложении.

При разработке веб-приложения с интерфейсами для регистрации и входа был применён веб-фреймворк Flask. При этом взаимодействие с базой данных происходит через PyMySql. Базы данных для каждого

алгоритма хеширования (Argon2, Scrypt и Bcrypt) хранят информацию о пользователях, их именах и хешах паролей.

Основные функции разработанного веб-приложения:

– get_db_connection: осуществляет подключение к базе данных;

– register_user: регистрирует пользователя, генерируя хеш для пароля с использованием всех трех алгоритмов и сохраняет данные в базы данных;

– authenticate_user: проверяет введенные данные для аутентификации;

– compare_hashing_algorithms: сравнивает время выполнения каждого алгоритма хеширования.

В процессе работы данного веб-приложения были использованы три библиотеки Python с функциями хеширования Argon2, Scrypt, Bcrypt и получены их средние времена выполнений хешированиий (на 100 попыток).

Экспериментальная часть

Экспериментальная часть исследования включала сравнение быстродействия алгоритмов генераций 100 хешей для каждого алгоритма с использованием библиотек:

– argon2-cffi для алгоритма Argon2;

– scrypt для алгоритма Scrypt;

– bcrypt для алгоритма Bcrypt.

Использовался компьютер с процессором Intel Core i7-10750H, 16 ГБ ОЗУ, ОС Ubuntu 22.04.

Для оценки производительности измерялось среднее время хеширования, а также анализировалась устойчивость к атакам через параметры памяти и вычислительной сложности.

Результаты

Результаты выполненного авторами эксперимента по сравнению быстродействия алгоритмов хеширования Argon2, Scrypt и Bcrypt в веб-приложении представлены в таблице 2.

Таблица № 2

Результаты сравнения быстродействия алгоритмов хеширования Argon2, Scrypt и Bcrypt в веб-приложении

Алгоритм хеширования	Среднее время выполнения хеширования	Количество генераций хешей
Argon2	73 мс	100
Scrypt	53 мс	100
Bcrypt	214 мс	100

В результате выполненного сравнения установлено, что наиболее объективные значения показателей выдала библиотека Python с функцией хеширования Scrypt, поскольку, помимо наименьшего среднего времени выполнения хеширования, библиотека так же позволяет добавить соль для хеширования, в то время как остальные библиотеки генерируют её самостоятельно.

Заключение

Алгоритм Bcrypt показал наименьшую производительность (при средней продолжительности хеширования, равной 214 мс) из-за фиксированных параметров памяти.

Scrypt продемонстрировал оптимальное сочетание времени выполнения хеширования и устойчивости к ASIC-атакам.

Argon2 обеспечил высокий уровень безопасности за счёт настраиваемых параметров, но уступил алгоритму Scrypt в быстродействии.

Полученные результаты согласуются с исследованиями [9-10], где Scrypt выделяется как эффективное решение для систем с высокими требованиями к производительности. Преимущество Argon2 в гибкости настройки делает его предпочтительным для приложений, где безопасность превалирует над скоростью. Ограничения Bcrypt, связанные с фиксированной памятью, свидетельствуют о необходимости его модернизации или замены в современных системах. Важным аспектом остается интеграция соли, которая в Scrypt реализована явно, что повышает устойчивость к радужным таблицам.

Литература

1. Качалкова С.В., Садыков А.М., Касимова А.Р. Механизм создания системы защиты информации при разработке распределённого центра обработки данных // Вестник Технологического университета. – 2018. – Т. 21, № 12. – С. 176-180.
2. Иванов М.А., Козырский Б.Л., Комаров Т.И., Копылова Д.С., Курышева О.К., Смирнов А.С., Спиридонов А.А. Хеш-функции: основы теории, новые конструкции и новые стандарты // REDS: Телекоммуникационные устройства и системы. – 2014. – Т. 4, № 3. – С. 285-288.
3. Мебония М.А., Федорова О.В. Сравнительное исследование хэш-алгоритмов в криптографии // Вестник науки. – 2022. – Т. 3, № 12(57). – С. 439-443.
4. Старанцова Е.В. Методы хеширования паролей // Пути инновационного развития науки и образования в современных условиях. – 2023. – С. 387-394.

5. Бушуев Р.А., Марченко А.В. Обзор алгоритмов хеширования на персональном компьютере // Политехнический молодежный журнал. – 2020. – № 3(44). – С. 2.
6. Гончаров А.М. Алгоритм Bcrypt как метод защиты от атак на данные аутентификации пользователя // Проблемы информационной безопасности социально-экономических систем. – 2022. – С. 87-88.
7. Родин А.А., Воронин В.А. Двухфакторная аутентификация как средство защиты учётной записи // Новые горизонты. – 2024. – С. 463-465.
8. Прокопов Д.С. Защита информации. Виды защиты информации // Передовые инновационные разработки. Перспективы и опыт использования, проблемы внедрения в производство – 2019. – С. 133-134.
9. Bromberg, L. Cryptographic Hash Functions and Some Applications to Information Security // Springer Proceedings in Mathematics and Statistics. – 2017. – P. 85-97.
10. Kundu, R. Cryptographic Hash Functions and Attacks – A Detailed Study // International Journal of Advanced Research in Computer Science. – 2020. – Vol. 11, No. 2. – P. 37-44.

References

1. Kachalkova S.V., Sadykov A.M., Kasimova A.R. Vestnik Texnologicheskogo universiteta. 2018. T. 21. № 12. pp. 176-180.
2. Ivanov M.A., Kozyrskij B.L., Komarov T.I., Kopylova D.S., Kurysheva O.K., Smirnov A.S., Spiridonov A.A. REDS: Telekommunikacionnye ustrojstva i sistemy. 2014. T. 4. № 3. pp. 285-288.
3. Meboniya M.A., Fedorova O.V. Vestnik nauki. 2022. T. 3. № 12(57). pp. 439-443.

4. Staranczova E.V. Puti innovacionnogo razvitiya nauki i obrazovaniya v sovremennyx usloviyax. 2023. pp. 387-394.
5. Bushuev R.A., Marchenko A.V. Politexnicheskij molodezhnyj zhurnal. 2020. № 3(44). p. 2.
6. Goncharov A.M. Problemy informacionnoj bezopasnosti social`no-ekonomicheskix sistem. 2022. pp. 87-88.
7. Rodin A.A., Voronin V.A. Novye gorizonty. 2024. pp. 463-465.
8. Prokopov D.S. Peredovye innovacionnye razrabotki. Perspektivy i opyt ispol`zovaniya, problemy vnedreniya v proizvodstvo. 2019. pp. 133-134.
9. Bromberg, L. Springer Proceedings in Mathematics and Statistics. 2017. pp. 85-97.
10. Kundu, R. International Journal of Advanced Research in Computer Science. 2020. vol. 11, No. 2. pp. 37-44.

Дата поступления: 9.06.2025
Дата публикации: 25.07.2025