УДК 004.056.5

Анализ современных методов криптографического шифрования и проектирование менеджера паролей с усиленной защитой учетных данных

Глухарев Михаил Леонидович — канд. техн. наук, доцент кафедры «Информатика и информационная безопасность». Научные интересы: информационная безопасность, криптографические методы защиты данных, проектирование защищенных программных систем, технологии защиты персональных данных. E-mail: mlgluharev@yandex.ru

Данилова Полина Ивановна — студент 4-го курса направления 10.05.03 «Информационная безопасность автоматизированных систем». Научные интересы: криптография, разработка приложений в сфере информационной безопасности, реализация защищенных пользовательских интерфейсов и алгоритмов шифрования. E-mail: polina_2004.11@mail.ru

Петербургский государственный университет путей сообщения Императора Александра I, Россия, 190031, Санкт-Петербург, Московский пр., 9

Для цитирования: Глухарев М. Л., Данилова П. И. Анализ современных методов криптографического шифрования и проектирование менеджера паролей с усиленной защитой учетных данных // Интеллектуальные технологии на транспорте. 2025. № 3 (43). С. 5–14. DOI: 10.20295/2413-2527-2025-343-5-14

Аннотация. *Представлено исследование о проектировании менеджера паролей с усиленной криптографической защитой для локального хранения учетных данных.* ***Цель:*** *создание программного решения, реализующего безопасную генерацию, шифрование и хранение паролей с использованием современных алгоритмов.* ***Методы:*** *применены современные криптографические подходы, включая симметричное шифрование (AES-GCM), хеш-функции (SHA-256), а также алгоритм PBKDF2 для генерации ключей на основе мастер-пароля и соли.* ***Результаты:*** *включают реализацию программного комплекса на Java, обеспечивающего настройку политики безопасности, защиту от атак перебором и словарных атак, а также проверку целостности данных. Разработан криптографический модуль, соответствующий требованиям NIST и ГОСТ.* ***Практическая значимость:*** *заключается в возможности применения программного продукта в организациях, в которых необходима локальная защита конфиденциальных данных без использования облачных технологий. Решение может использоваться для повышения уровня информационной безопасности в малом бизнесе, образовательных учреждениях и в среде разработчиков.* ***Обсуждение:*** *подчеркиваются преимущества автономной архитектуры, открытого кода и перспективы дальнейшего развития, включая расширение функционала и адаптацию для корпоративных нужд.*

Ключевые слова: *менеджер паролей, криптографическая защита, AES-GCM, PBKDF2, SHA-256, хранение учетных данных, шифрование, информационная безопасность, Java*

2.3.6 *— методы и системы защиты информации, информационная безопасность (технические науки)*

Введение

С развитием цифровых технологий значительно возросло количество сервисов, требующих регистрации и хранения учетных данных. Пользователи зачастую используют слабые пароли или один и тот же пароль на различных платформах, что приводит к риску утечки данных. Согласно отчету Positive Technologies [1], более половины (52 %) успешных атак на организации во втором полугодии 2024 года закончились утечками данных. Наиболее часто жертвами утечек становились госучреждения (13 %), промышленность (10 %), IT-компании (10 %), финансовые организации (8 %) и медицинские учреждения (7 %).

Менеджеры паролей являются важным инструментом для обеспечения защиты от несанкционированного доступа и криптостойкости зашифрованных данных. В статье представлена разработка менеджера паролей с поддержкой криптографической защиты, реализованного на языке Java. Программа включает в себя функционал генерации паролей, управления политиками безопасности и шифрования данных с использованием алгоритмов AES-256 и PBKDF2.

Анализ существующих решений и постановка задачи

Существует значительное количество программных решений, предназначенных для хранения и управления паролями, среди которых наиболее популярными являются KeePass [2], Bitwarden [3], LastPass [4] и другие. Эти системы предоставляют пользователям возможность централизованного хранения учетных данных, автоматической генерации паролей, а также удобные средства для передачи и синхронизации информации между различными устройствами и платформами. Вместе с тем анализ существующих программных средств выявляет ряд важных ограничений, оказывающих влияние на уровень доверия к таким решениям и их применимость в контексте задач, связанных с повышенными требованиями к безопасности.

Во-первых, значительная часть широко распространенных менеджеров паролей разрабатывается как проприетарное программное обеспечение, что не дает возможности специалистам по информационной безопасности провести анализ шифрования данных и убедиться в отсутствии уязвимостей, ошибок или закладок.

Во-вторых, большинство современных сервисов ориентированы на облачную модель хранения: пользовательские данные сохраняются на внешних серверах поставщика услуги, даже если они предварительно зашифрованы. Такая архитектура порождает дополнительные риски: в случае компрометации серверной инфраструктуры, утечки административных ключей или нарушения политики безопасности со стороны персонала возможно получение злоумышленником доступа ко всем хранилищам пользователей. Подобные инциденты уже имели место в реальной практике, например в случае взлома LastPass в 2022 году [5], что подтверждает уязвимость облачных решений перед целенаправленными атаками.

Кроме того, большинство программных решений не позволяют пользователю гибко настраивать политику генерации паролей. Такой подход может быть неприемлем в образовательной, исследовательской или корпоративной среде, где важно обеспечить не только безопасность, но и возможность контроля формирования данных.

Учитывая изложенное, целесообразным является создание локального менеджера паролей с открытой архитектурой, реализующего принципы автономной защиты и криптографической стойкости алгоритмов шифрования. В данном случае пользователю не предоставляется полный контроль над процессом обработки данных, но доступность открытого кода обеспечивает прозрачность и возможность проверки безопасности на всех этапах — от генерации ключа на основе мастер-пароля до анализа алгоритма шифрования и модификации кода.

В работе представлена реализация данного решения на языке Java, ориентированная на применение современных криптографических стандартов и обеспечение высокого уровня безопасности при локальной работе без необходимости обращения к внешним серверам.

Функционал программы

Разработанный менеджер паролей позволяет создавать и безопасно хранить учетные данные с учетом заданной политики безопасности (рис. 1).

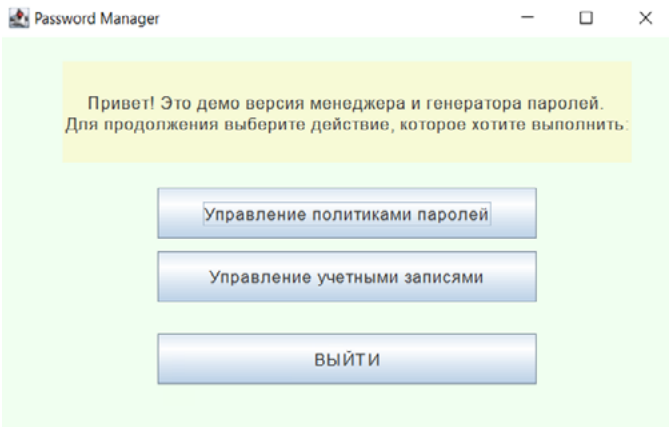


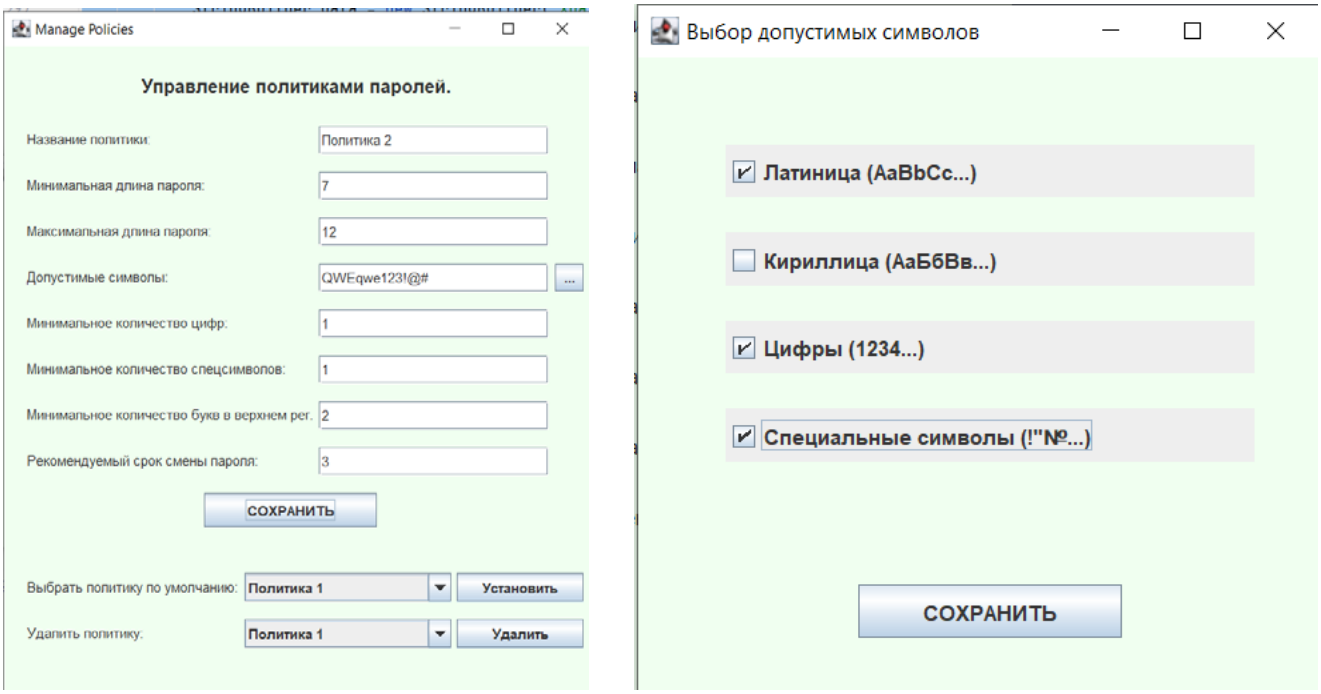
Рис. 1. Главный экран программы

Генерация паролей происходит автоматически на основе заранее определенных требований: параметры длины (минимальной и максимальной), допустимого набора символов, а также требований к числу цифр, специальных символов и символов верхнего регистра. Пользователь может настраивать эти правила, создавать и удалять их, а также выбирать одну из политик в качестве основной (рис. 2).

Все учетные данные, включая логины, пароли и дополнительную информацию (например, описание сервиса, дату добавления и т. п.) после сохранения (рис. 3), подвергаются обязательному шифрованию перед сохранением в отдельном файле.

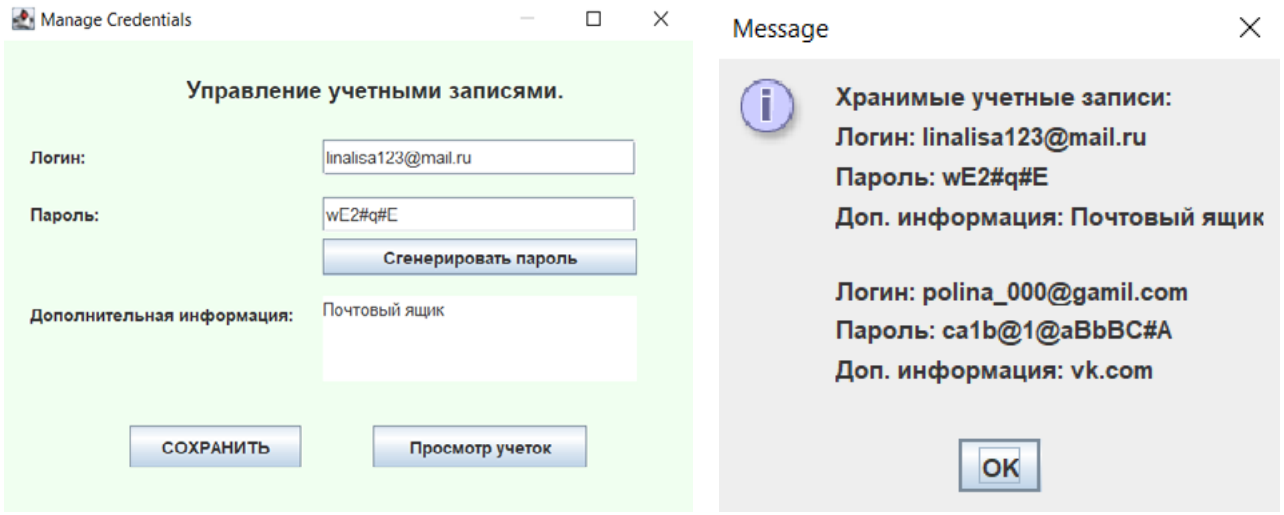
В качестве криптографической основы используется симметричный алгоритм AES (Advanced Encryption Standard) в режиме GCM (Galois/Counter Mode), который гарантирует не только конфиденциальность, но и целостность хранимой информации. Ключ шифрования формируется с применением функции PBKDF2 (Password-Based Key Derivation Function 2) на основе введенного пользователем мастер-пароля и случайной последовательности байтов, что обеспечивает стойкость к атакам перебором и словарным атакам.

Доступ к зашифрованным данным возможен только после ввода правильного мастер-пароля (рис. 4). При этом проверка пароля осуществляется путем сравнения его хэш-значения с ранее сохраненным эталонным значением, сгенерированным по алгоритму SHA-256.



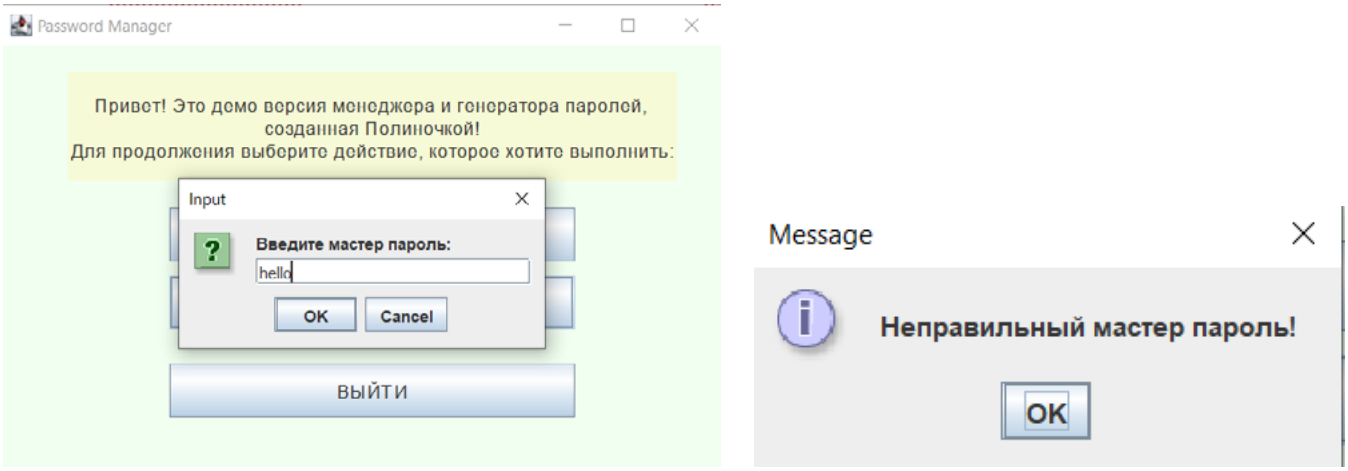
а *б*

Рис. 2. Управление политиками паролей:
а — настройка общих правил; *б* — выбор допустимых символов



а *б*

Рис. 3. Управление учетными записями
а — сохранение новых записей; *б* — просмотр существующих учетных записей



а *б*

Рис. 4. Ввод мастер-пароля:
а — ввод пароля; *б* — сообщение о неправильно введенном пароле

Такой подход исключает возможность несанкционированного доступа даже в случае получения доступа к файлу с хранимыми данными.

Общая схема криптографической защиты данных

Современные криптографические системы строятся с учетом трех фундаментальных угроз информационной безопасности: утечка конфиденциальной информации, нарушение целостности и доступности данных. Нарушение конфиденциальности выражается в получении злоумышленником доступа к содержанию передаваемых или хранимых сообщений, нарушение целостности — в несанкционированном и незаметном для получателя изменении данных, нарушение доступности — действие, которое делает невозможным или затрудняет доступ к ресурсам информационной системы.

Дешифрование сообщения злоумышленником возможно в случае вычисления секретного ключа либо при обнаружении алгоритма, функционально эквивалентного используемому в системе и не требующего знания ключа. Таким образом, разработка криптографически устойчивых систем основывается на двух характеристиках: криптостойкости — способности защищать сообщения от расшифровки без знания ключа и имитостойко-

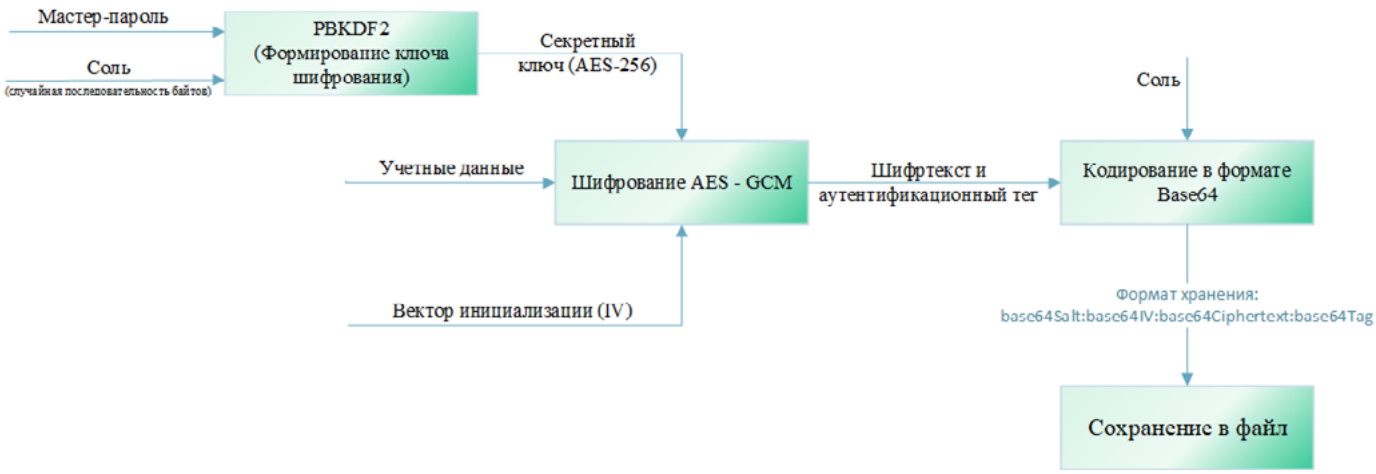


Рис. 5. Общая схема процесса криптографической защиты учетных данных

сти — устойчивости к попыткам генерации ложных данных, внешне идентичных подлинным [6].

Криптографическая защита учетных данных в разработанном программном обеспечении реализуется посредством применения симметричных алгоритмов шифрования и стандартных методов усиления паролей. Используемые подходы соответствуют актуальным требованиям международных и российских нормативных документов в области защиты информации, в том числе рекомендациям NIST [7] и ГОСТ Р ИСО/МЭК 27002—2021 [8].

На рис. 5 представлена общая схема криптографической защиты данных, начиная от ввода мастер-пароля и заканчивая формированием итогового файла с зашифрованной информацией.

Вначале мастер-пароль пользователя и случайно сгенерированная последовательность битов (соль) поступают на вход алгоритма PBKDF2, описанного в стандарте RFC 8018 [9], где на основе алгоритма HMAC-SHA256 выполняется итеративное хеширование. Это позволяет значительно повысить вычислительную сложность атаки полным перебором [6]. Результатом выполнения PBKDF2 является криптографически стойкий ключ длиной 256 бит, используемый в последующем симметричном шифровании.

В качестве алгоритма шифрования был выбран стандарт AES в режиме GCM. Помимо шифрования записей, режим GCM обеспечивает аутентификацию данных за счет формирования специального аутентификационного тега.

В результате работы системы на выходе модуля шифрования формируются зашифрованный текст и тег аутентификации, которые вместе с вектором инициализации и солью преобразуются в формат Base64 и сохраняются в файл. Такая структура хранения данных обеспечивает криптографическую стойкость, не усложняя при этом процесс обращения к данным.

Генерация ключа шифрования: применение PBKDF2 с HMAC-SHA256

Современные алгоритмы блочного шифрования, включая AES, используют ключ фиксированной длины — 128, 192 или 256 бит. Из него формируются подключи, используемые на каждом раунде. Существуют специализированные криптоаналитические атаки, которые нацелены именно на алгоритмы генерации подключей. В связи с этим одним из ключевых требований к конструкции блочных шифров является наличие надежного алгоритма (или процедуры) расширения ключа. Усложнение алгоритма формирования расширенного ключа может повысить общую стойкость шифра [10]. Для этого было принято решение использовать функцию PBKDF2. Блок-схема алгоритма представлена на рис. 6.

Механизм, представленный на рис. 6, реализует процесс усложнения пароля на основе итеративной хеш-функции и логической операции «исключающее или». На первом этапе вводимый пользователем мастер-пароль преобразуется в битовую последовательность и подвергается хешированию

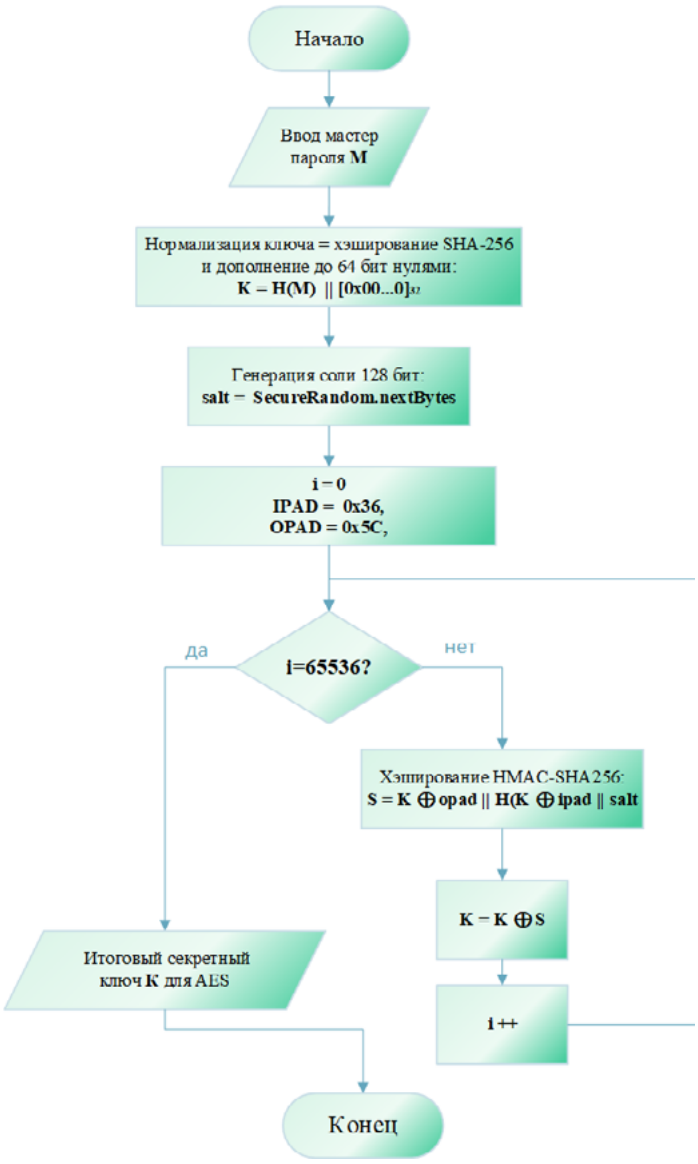


Рис. 6. Блок-схема алгоритма формирования расширенного криптографического ключа

функцией SHA-256 [6]. Далее формируется соль — случайная строка длиной 128 бит, уникальная для каждой сохраненной записи, и добавляется к полученному хешу.

На следующем этапе используется механизм HMAC (Hash-Based Message Authentication Code), который объединяет мастер-пароль и соль, формируя начальное значение хеша. Далее выполняется последовательность итераций, где каждая итерация HMAC принимает в качестве входных значений результат вычислений предыдущего этапа. Полученные на каждой итерации хеш-значения накапливаются с помощью побитовой операции XOR, что обеспечивает постепенное усложнение выходного ключа и снижает возможность его предсказания.

Чем больше итераций задано, тем дольше будет длиться процесс генерации ключа, что создает вычислительные сложности для атакующего, пытающегося перебрать возможные значения пароля. Такая схема эффективно замедляет атаки грубой силы (brute-force) и делает неэффективными словарные атаки, особенно в условиях, когда ресурсы злоумышленника ограничены во времени или мощности.

Итоговым результатом работы алгоритма становится ключ длиной 256 бит, применяемый в блочном алгоритме AES. Важно отметить, что каждая новая сохраняемая учетная запись будет иметь новую последовательность битов в соли. Это означает, что даже если два пользователя используют одинаковые пароли, их хеши будут разными из-за уникальной соли. Это делает атаки с использованием радужных таблиц (rainbow tables) менее эффективными, так как злоумышленник не может заранее подготовить таблицы хешей [11].

Применение алгоритма шифрования AES в режиме GCM

AES-GCM — вариант шифра AES в режиме гаммирования с аутентификацией Галуа. Алгоритм генерирует аутентификационный тег на основе открытого текста, дополнительных аутентифицированных данных и ключа аутентификации [12].

Подробный порядок выполнения операций отображен на рис. 7.

На первом этапе генерируется вектор инициализации фиксированной длины (96 бит), который используется в качестве начального значения для счетчика J0. Далее исходные данные разбиваются на блоки по 128 бит, каждый из которых подвергается шифрованию при помощи алгоритма AES с ключом длиной 256 бит. Полученные зашифрованные блоки объединяются с блоками открытого текста путем выполнения побитовой операции XOR.

Параллельно с этим осуществляется вычисление тега аутентификации. Для этого используется функция GHASH, использующая умножение в

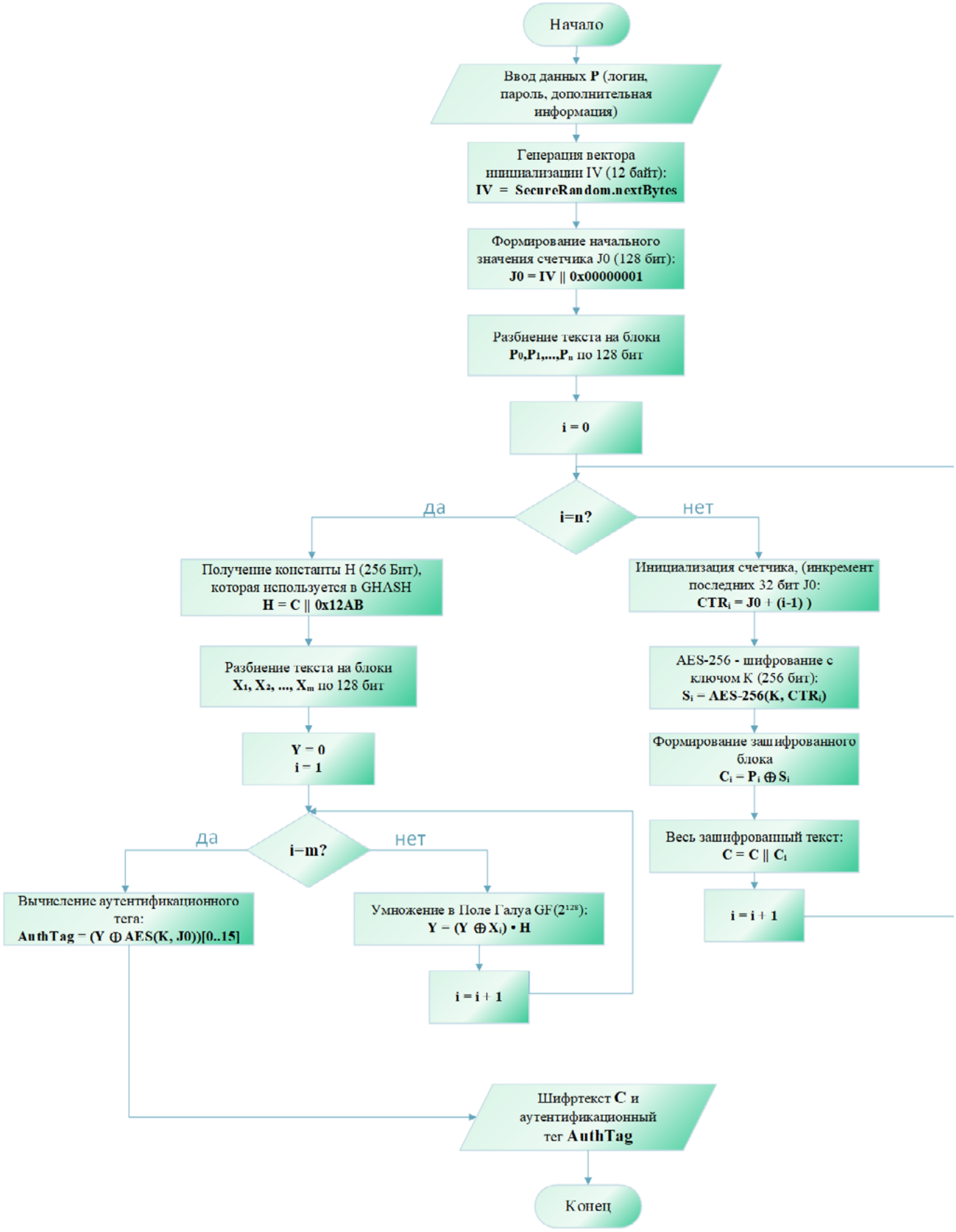


Рис. 7. Блок-схема шифрования AES-GCM

поле Галуа GF(2^{128}) [13]. Она последовательно обрабатывает все блоки данных и служебные параметры, включая вектор инициализации и длину сообщения. В результате формируется тег, который позволяет при расшифровке проверить целостность зашифрованной информации.

Зашифрованное сообщение, вектор инициализации, соль и тег аутентификации объединяются и кодируются в формате Base64. Полученная структура данных сохраняется в файл в виде строки, обеспечивая удобство хранения.

Заключение

Реализуемое решение основывается на применении проверенных и широко признанных криптографических стандартов. В частности, алгоритм PBKDF2 используется для получения криптографического ключа из мастер-пароля с использованием криптографической соли и большого количества итераций, что существенно повышает устойчивость к атакам методом перебора. Для обеспечения конфиденциальности и аутентификации данных применяется алгоритм симметричного шифрования AES в режиме GCM, обладающий встроенным механизмом формирования аутентификационных тегов. В качестве хеш-функции используется SHA-256, обладающая высокой криптостойкостью и обеспечивающая контроль целостности обрабатываемых данных.

Такой подход соответствует современным стандартам информационной безопасности, защищая данные как от несанкционированного доступа (за счет шифрования), так и от нарушения целостности (благодаря аутентификации данных).

Разработанная программа ориентирована на учебно-прикладное использование и может быть полезна в рамках курсов, связанных с программированием, криптографией и информационной безопасностью. Она подходит для индивидуального использования студентами и преподавателями, а также может служить демонстрационной платформой для освоения принципов шифрования, генерации ключей и управления учетными записями. Программа также может применяться для защиты личных данных на персональных устройствах в условиях локального доступа без подключения к сети.

СПИСОК ИСТОЧНИКОВ

1. Авезова Я., Рыжков В. Утечки конфиденциальных данных из организаций: второе полугодие 2024 года // Positive Technologies. 2025. 13 марта. URL: http://www.ptsecurity.com/ru-ru/research/analytics/utechki-dannyh-aktualnye-ugrozy-vtorogo-polugodiya-2024-dlya-organizaczij (дата обращения: 05.07.2025).

2. KeePass Password Safe. URL: http://keepass.info (дата обращения: 06.07.2025).

3. Bitwarden Password Manager. URL: http://bitwarden.com (дата обращения: 06.07.2025).

4. LastPass. URL: http://lastpass.com (дата обращения: 06.07.2025).

5. Toubba K. 12-22-2022: Notice of Security Incident LastPass // LastPass Blog. 2022. 22 December. URL: http://blog.lastpass.com/posts/notice-of-security-incident (дата обращения: 05.07.2025).

6. Иванов М. А., Чугунков И. В. Криптографические методы защиты информации в компьютерных системах и сетях: учебное пособие / под ред. М. А. Иванова. М.: НИЯУ МИФИ, 2012. 400 с.

7. National Institute of Standards and Technology. Computer Security Resource Center. Publications Database. URL: http://csrc.nist.gov/publications (дата обращения: 05.07.2025).

8. ГОСТ Р ИСО/МЭК 27002—2021. Информационные технологии. Методы и средства обеспечения безопасности. Свод норм и правил применения мер обеспечения информационной безопасности = Information technology. Security techniques. Code of practice for information security controls: национальный стандарт Российской Федерации: введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 20 мая 2021 г. № 416-ст: дата введения: 2021-11-30. М.: Стандартинформ, 2021. 74 с.

9. Moriarty K., Kaliski B., Rusch A. PKCS #5: Password-Based Cryptography Specification Version 2.1 (RFC 8018). 2017. 40 p. URL: https://datatracker.ietf.org/doc/html/rfc8018 (дата обращения: 05.07.2025).

10. Молдовян Н. А., Молдовян А. А., Еремеев М. А. Криптография: от примитивов к синтезу алгоритмов. СПб.: БХВ-Петербург, 2004. 448 с.

11. Баланов А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами: учебное пособие для вузов. 2-е изд., стер. СПб.: Лань, 2025. 312 с.

12. Хлебников А. OpenSSL 3. Ключ к тайнам криптографии. Лучшие способы повысить безопасность сети с применением OpenSSL 3 = Demystifying Criptography with OpenSSL 3.0. Discover the best techniques to enhance your network seciruty with OpenSSL 3.0 / пер.с англ. А. А. Слинкина. М.: ДМК Пресс, 2023. 300 с.

13. Каширская Е. Н., Кушнир А. П. Криптографические системы: учебное пособие. М.: МИРЭА — Российский технологический университет, 2021. 66 с.

Дата поступления: 13.07.2025
Решение о публикации: 06.08.2025

Analysis of Modern Cryptographic Encryption Methods and Design of a Password Manager with Enhanced Credential Protection

Mikhail L. Glukharev — PhD in Engineering, Associate Professor of the Information Technology and IT Security Department. Research interests: information security, cryptographic data protection methods, design of secure software systems, technologies for personal data protection.
E-mail: mlgluharev@yandex.ru

Polina I. Danilova — 4th year Specialist's Degree Student in 10.05.03 Information Security of Automated Systems specialism. Research interests: cryptography, development of applications in the field of information security, implementation of secure user interfaces and encryption algorithms.
E-mail: polina_2004.11@mail.ru

Emperor Alexander I St. Petersburg State Transport University, 9, Moskovsky ave., Saint Petersburg, 190031, Russia

For citation: Glukharev M. L., Danilova P. I. Analysis of Modern Cryptographic Encryption Methods and Design of a Password Manager with Enhanced Credential Protection. *Intellectual Technologies on Transport*, 2025, No. 3 (43), Pp. 5–14. DOI: 10.20295/2413-2527-2025-343-5-14. (In Russian)

Abstract. *The study presents the development of a password manager with enhanced cryptographic protection for local storage of credentials.* ***Purpose:*** *to create a software solution that implements secure password generation, encryption, and storage using modern algorithms.* ***Methods:*** *modern cryptographic approaches were applied, including symmetric encryption (AES-GCM), hash functions (SHA-256), and the PBKDF2 algorithm for key generation based on a master password and salt.* ***Results:*** *the implementation of a Java-based software package that allows password policy configuration, protection against brute-force and dictionary attacks, and data integrity verification. A cryptographic module compliant with NIST and GOST standards has been developed.* ***Practical significance:*** *the software can be used in organizations requiring local protection of confidential data without relying on cloud technologies. It will be suitable for improving information security in small businesses, educational institutions, and development environments.* ***Discussion:*** *the paper highlights the advantages of an autonomous architecture, open-source code, and prospects for further development, including functionality expansion and adaptation for corporate use.*

Keywords: *password manager, cryptographic protection, AES-GCM, PBKDF2, SHA-256, credential storage, encryption, information security, Java*

REFERENCES

1. Avezova Ya., Ryzhkov V. Utechki konfidentsialnykh dannykh iz organizatsiy: vtoroe polugodie 2024 goda [Leaks of confidential data from organizations: the second half of 2024], *Positive Technologies*. Published online at March 13, 2025. Available at: http://www.ptsecurity.com/ru-ru/research/analytics/utechki-dannyh-aktualnye-ugrozy-vtorogo-polugodiya-2024-dlya-organizaczij (accessed: July 05, 2025). (In Russian).

2. KeePass Password Safe. Available at: http://keepass.info (accessed: July 06, 2025).

3. Bitwarden Password Manager. Available at: http://bitwarden.com (accessed: July 06, 2025).

4. LastPass. Available at: http://lastpass.com (accessed: July 06, 2025).

5. Toubba K. 12-22-2022: Notice of Security Incident LastPass, *LastPass Blog*. Published online at December 22, 2022. Available at: http://blog.lastpass.com/posts/notice-of-security-incident (accessed: July 05, 2025).

6. Ivanov M. A., Chugunkov I. V. Kriptograficheskie metody zashchity informatsii v kompyuternykh sistemakh i setyakh: uchebnoe posobie [Cryptographic methods of information protection in computer systems and networks: tutorial]. Moscow, National Research Nuclear University MePhI, 2012, 400 p. (In Russian)

7. National Institute of Standards and Technology. Computer Security Resource Center. Publications Database. Available at: http://csrc.nist.gov/publications (accessed: July 05, 2025).

8. GOST R ISO/MEK 27002—2021. Informatsionnye tekhnologii. Metody i sredstva obespecheniya bezopasnosti. Svod norm i pravil primeneniya mer obespecheniya informatsionnoy bezopasnosti [GOST R ISO/IEC 27002—2021. Information technology. Security techniques. Code of practice for information security controls]. Effective from November 30, 2021. Moscow, StandartInform Publishing House, 2021, 74 p. (In Russian)

9. Moriarty K., Kaliski B., Rusch A. PKCS #5: Password-Based Cryptography Specification Version 2.1 (RFC 8018). 2017, 40 p. Available at: https://datatracker.ietf.org/doc/html/rfc8018 (accessed: July 05, 2025).

10. Moldovyan N. A., Moldovyan A. A., Eremeev M. A. Kriptografiya: ot primitivov k sintezu algoritmov [Cryptography: from primitives to the synthesis of algorithms]. Saint Petersburg, BHV-Peterburg Publishing House, 2004, 448 p. (In Russian)

11. Balanov A. N. Bekend-razrabotka veb-prilozheniy: arkhitektura, proektirovanie i upravlenie proektami: uchebnoe posobie dlya vuzov [Backend development of web applications: architecture, design and project management: a tutorial for universities]. Saint Petersburg, LAN Publishing House, 2025, 312 p. (In Russian)

12. Khlebnikov A. OpenSSL 3. Klyuch k taynam kriptografii. Luchshie sposoby povysit bezopasnost seti s primeneniem OpenSSL 3 [Demystifying Criptography with OpenSSL 3.0. Discover the best techniques to enhance your network seciruty with OpenSSL 3.0]. Moscow, DMK Press Publishing House, 2023, 300 p. (In Russian)

13. Kashirskaya E. N., Kushnir A. P. Kriptograficheskie sistemy: uchebnoe posobie [Cryptographic systems: a tutorial]. Moscow, MIREA — Russian Technological University, 2021, 66 p. (In Russian)

Received: 13.07.2025
Accepted: 06.08.2025