УДК 519

ПРОГРАММНАЯ РАЗРАБОТКА МЕНЕДЖЕРА ПАРОЛЕЙ С ИСПОЛЬЗОВАНИЕМ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ ЗАЩИТЫ ДАННЫХ

Н. В. Егоров, К. А. Сенько, О. А. Сафарьян
Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

В статье проанализированы основные принципы работы наиболее используемых менеджеров паролей — программных средств, позволяющих хранить информацию о паролях в безопасном виде. На основе проведенного анализа были изучены проблемы, содержащиеся в рассмотренных программных реализациях. Результатом работы явилось разработанное программное средство, используемое для безопасного хранения паролей.

Ключевые слова: менеджер паролей, криптографические алгоритмы защиты информации, хеш-функция, AES, GCM, SHA256.

SOFTWARE DEVELOPMENT OF PASSWORD MANAGER USING CRYPTOGRAPHIC ALGORITHMS OF DATA PROTECTION.

N. V. Egorov, K. A. Senko, O. A. Safaryan
Don State Technical University (Rostov-on-Don, Russian Federation)

This article analyzes the basic principles of the most popular password managers – application software for keeping information about passwords in safe. Based on the analysis, the problems in the considered software implementations were studied. The result of the work was a developed software tool used for secure password storage.

Keywords: password manager, cryptographic algorithms of data protection, hash-functions, AES, GCM, SHA256.

Введение. На сегодняшний день всё большее количество различных сервисов предоставляют свои услуги с использованием сети интернет. Для использования этих услуг пользователям требуется производить аутентификацию с помощью паролей — данных, знание которых подтверждает личность пользователя на онлайн ресурсе.

Для увеличения безопасности сохранности личных данных пользователей (ФИО, адрес, контактная информация и т.д.) многие онлайн сервисы требуют использования паролей, которые имеют определённую длину (не менее 10–12 знаков), содержат заглавные и строчные буквы, цифры и специальные символы (например, ~!@#$%^&*()-+{}";:.,/) [1]. Данный факт делает процесс запоминания паролей или их хранение на бумажных носителях достаточно сложной задачей.

Основная часть. Для безопасного хранения паролей на цифровых устройствах используются программные средства — менеджеры паролей. Эти средства используют криптографические методы защиты информации для безопасного хранения данных и представляют их в случае необходимости [2].

Хотя потребность использования менеджеров паролей на сегодняшний день высокая, многие пользователи испытывают оправданное недоверие к предоставлению своей информации программным средствам различных компаний в связи с множественными сообщениями об утечке или продаже информации. Поэтому целями данной статьи являются:

– проведение анализа основных принципов работы менеджеров паролей;

– реализация программного средства для безопасного хранения паролей.

Аналитический обзор. Рассмотрим наиболее популярные программные средства, используемые для безопасного хранения данных пользователей.

Dashlane — менеджер паролей, позволяющий производить хранение паролей на облачном хранилище. Встроенный в данное программное средство VPN модуль позволяет производить обмен информацией с использованием криптографических алгоритмов. Так как хранение данных производится на удалённом хранилище, то можно заметить, что без наличия у пользователя доступа к сети, получение хранящихся данных будет невозможно.

1Password — программное средство, позволяющее хранить конфиденциальную информацию (пароли, данные банковских карт и т. д.). Для доступа к сохранённой информации используется мастер-пароль пользователя. Безопасность хранимых данных обусловлена процессом их шифрования с использованием симметричного криптографического алгоритма AES. В отличии от программного средства Dashline, менеджер паролей 1Password позволяет выбрать место хранения данных — локальное или облачное хранилище.

Icloud Keychain — программное средство, предназначенное для хранения паролей на устройствах, производимых компанией Apple. Доступ к сохранённой информации используется при помощи средств аутентификации владельца устройства: пин-код, мастер-пароль, биометрия (touch id, face id). Хранение данных происходит в зашифрованном виде с использованием симметричного алгоритма шифрования AES.

На основании обзора программных решений в сфере безопасного хранения паролей сформулируем необходимые требования к программному средству, которое будет в дальнейшем реализовано:

– возможность использования программного средства на различных видах операционных систем, таких как Windows, Linux и т.д.;

– локальное хранение данных для обеспечения постоянного доступа к ним пользователя;

– использование криптографических методов защиты информации для обеспечения безопасного хранения данных пользователя;

– возможность использования разных паролей для хранения различных данных, так как при разоблачении одного мастер-пароля злоумышленник может получить доступ ко всем хранимым данным пользователя.

Программная реализация. Для программной реализации в качестве языка разработки был выбран язык Golang [3], так как он обладает следующими характеристиками:

– строгая статическая типизация;

– автоматическое управление памятью с помощью использования сборщика мусора;

– высокая читаемость исходного кода программ;

– наличие средств обработки ошибок исполнения программы;

– стандартная библиотека, реализующая необходимый функционал.

Результатом работы программы, написанной на языке Golang, является бинарный файл, который не требует установки самого языка для исполнения (для запуска кода, написанного на языке Python необходимо наличие интерпретатора python) и, в зависимости от настроек процесса компиляции (GOOS — вид операционный системы, GOARCH — архитектура процессора), может быть использован на различных видах операционных систем [4].

Для безопасного хранения пользовательских данных в ходе работы реализуемого менеджера паролей использовались следующие библиотеки:

– crypto/aes — библиотека, содержащая реализацию симметричного алгоритма шифрования AES — Advanced Encryption Standard;

– crypto/cipher — библиотека, содержащая реализацию режима работы GCM — Galois Counter Mode;

– crypto/sha256 — библиотека, содержащая реализацию хеш-функции SHA256 — Secure Hash Algorithm.

Основной алгоритм шифрования, используемый в ходе работы реализуемого менеджера паролей — AES256 [6], который использует одинаковое значение ключа шифрования в процессах шифрования и расшифровки пользовательских данных. Для обеспечения безопасности используется режим шифрования GCM — Galois/Counter Mod (счётчик с аутентификацией Галуа), который позволяет добавлять случайные значения в ходе шифрования данных. Случайные значения изменяют финальное представление зашифрованных данных, что делает невозможным процесс распознавания исходных данных относительно зашифрованных.

Для генерации ключа, используемого в ходе работы алгоритма AES256, используется хеш-функция SHA256. В качестве аргумента данной функции используется последовательность, введённая пользователем для работы с менеджером паролей. Результатом работы хеш-функции SHA256 является последовательность из 32 байт, которая обладает следующими свойствами:

– минимальное изменение аргумента хеш-функции (изменение одного бита аргумента) приводит к сильному изменению результата;

– невозможно получить значение аргумента хеш-функции путём анализа результирующей последовательности байт;

– невозможно подобрать значение аргумента хеш-функции для получения известной последовательности байт.

На **Ошибка! Источник ссылки не найден.** и 2 представлены примеры использования разработанного менеджера пароля для шифрования данных в операционных системах Windows и Linux.

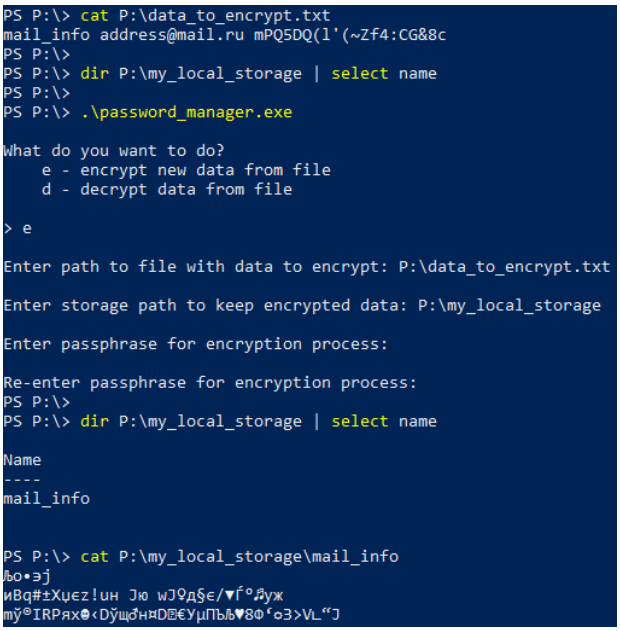

Рис. 1. Пример шифрования данных в ОС Windows

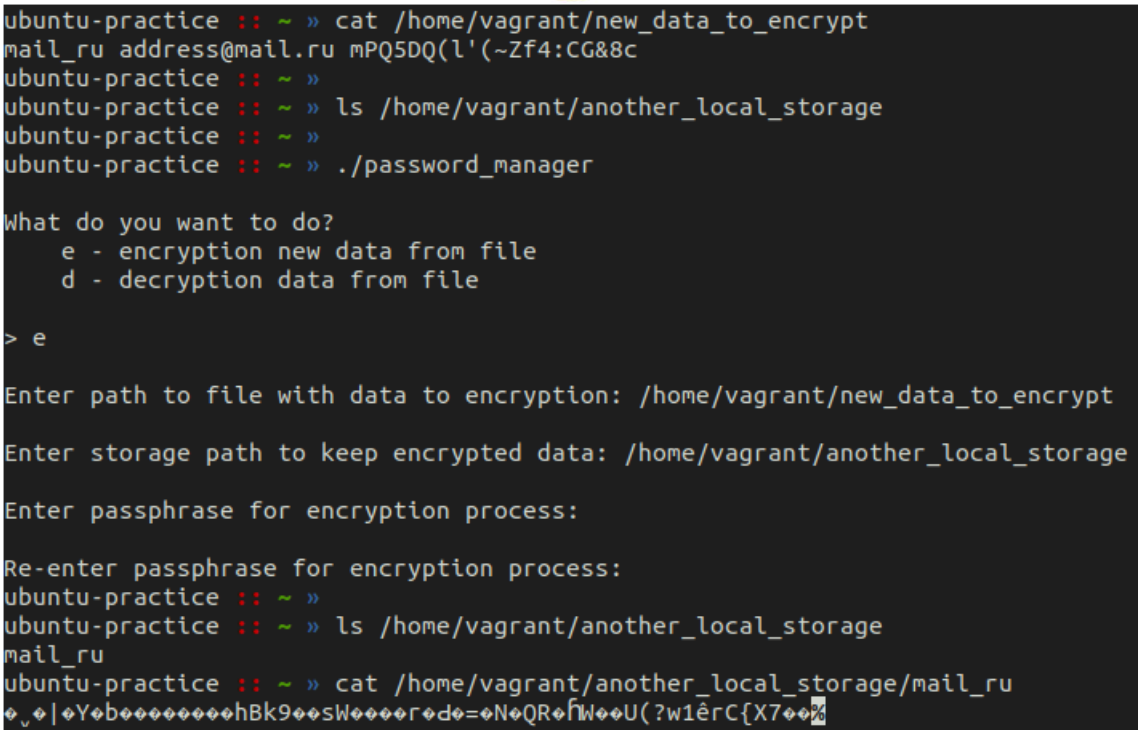

Рис. 2. Пример шифрования данных в ОС Linux

Как видно из показанного выше примера, ввод последовательности, используемой в процессе шифрования данных происходит дважды. Это необходимо для подтверждения правильности введённого. Вводимая пользователем последовательность не отображается для обеспечения сохранности данной последовательности в секрете.

На **Ошибка! Источник ссылки не найден.** и 4 представлены примеры использования разработанного менеджера паролей для расшифровки данных в операционных системах Windows и Linux.

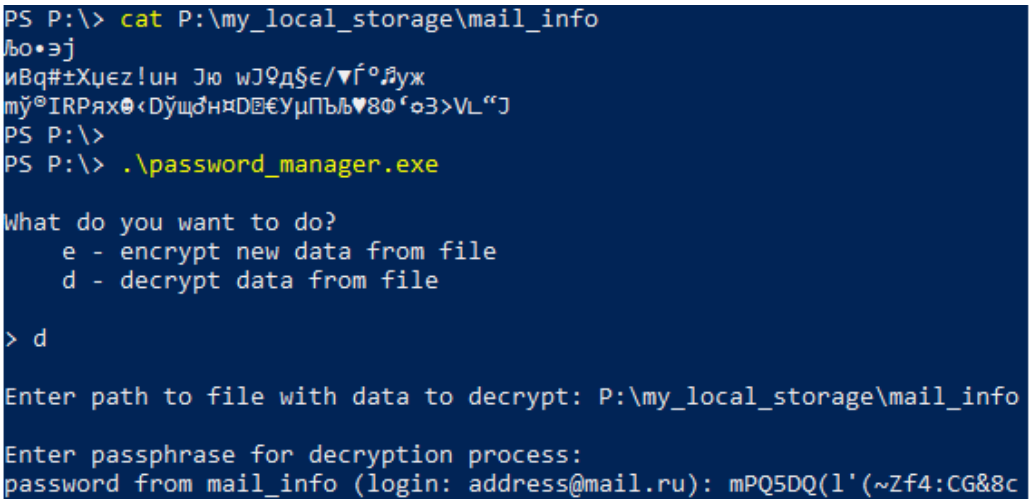

Рис. 3. Пример расшифровки данных на ОС Windows

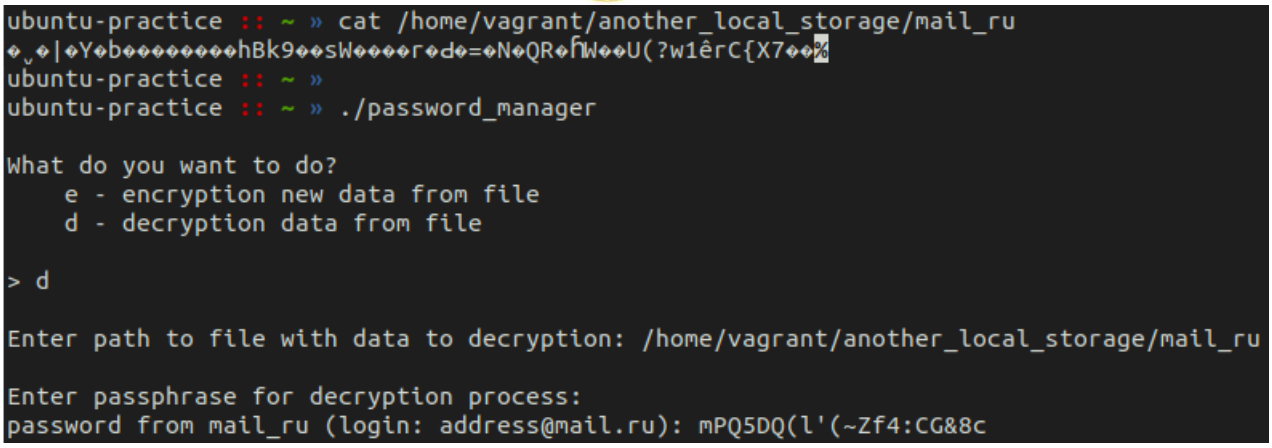

Рис. 4. Пример расшифровки данных на ОС Linux

Результатом работы менеджера паролей в процессе расшифровки данных является вывод сообщения с заданными логином и паролем.

Заключение. После анализа существующих средств, используемых для безопасного хранения паролей, был реализован и описан собственный менеджер паролей, который обеспечивает безопасное хранение информации. Плюсами данной программной реализации являются:

– возможность использования разработанного менеджера паролей на различных видах операционных систем;

– свойство локального хранения зашифрованной информации, обеспечивающее постоянный доступ к ней;

– возможность использования различных последовательностей вместо одного мастер-ключа, которые используются для шифрования и расшифровки данных;

– использование хеш-функции при создании ключа шифрования/расшифровки для увеличения безопасности зашифрованных данных к атаке полного перебора значений ключа криптографического алгоритма AES256.

Библиографический список

1. Как создать надежный пароль: подробное руководство / 10guards :[сайт]. — URL: https://10guards.com/ru/articles/how-to-create-a-strong- password-step-by-step-guide/ (дата обращения : 20.03.2022).

2. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы и исходный код на С / Б. Шнайер. — Москва : Диалектика, 2016. — 1024 с.

3. Официальный сайт языка программирования Golang / GO : [сайт]. — URL: https://go.dev (дата обращения : 21.03.2022).

4. Таненбаум, Э. Современные операционные системы / Э. Таненбаум — Питер : Классика Computer Science, 2018. — 920 с.

5. Омассон, Ж. О криптографии всерьез / Ж. Омассон. — Москва : ДМК Пресс, 2022. — 328 с.

Об авторах:

Егоров Никита Валентинович, студент кафедры «Компьютерная безопасность» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), deadpool.13_37@mail.ru

Сенько Кирилл Андреевич, студент кафедры «Компьютерная безопасность» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), tmpolo@mail.ru

Сафарьян Ольга Александровна, доцент кафедры «Компьютерная безопасность» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат технических наук, доцент, ORCID, safari_2006@mail.ru

About the Authors:

Egorov, Nikita V., Student, Department of Computer Security, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), deadpool.13_37@mail.ru

Senko, Kirill A., Student, Department of Computer Security, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), tmpolo@mail.ru

Safaryan, Olga A., Associate professor, Department of Computer Security, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), Cand.Sci. (Eng.), associate professor, ORCID, safari_2006@mail.ru