Pak. j. life soc. Sci. (2024), 22(2): 20107-20115 E-ISSN: 2221-7630;P-ISSN: 1727-4915

Pakistan Journal of Life and Social Sciences
www.pjlss.edu.pk

https://doi.org/10.57239/PJLSS-2024-22.2.001472

RESEARCH ARTICLE

Development of Password Manager Using Cryptographic Algorithms for Data Protection in Windows and Linux Operating Systems

L.V. Cherckesova[1*], O.A. Safaryan[2], O.S. Buryakova[3], V.M. Porksheyan[4]

[1,2,3,4] Don State Technical University, Rostov-on-Don, Russia

ARTICLE INFO

Received: Oct 26, 2024

Accepted: Dec 11, 2024

Keywords

Cryptography

Software Development

Data Protection

Cryptographic Algorithm

***Corresponding Author:**

chia2002@inbox.ru

ABSTRACT

This study aimed to develop a password manager using cryptographic algorithms to ensure secure data storage. The research drew on monographs, scientific articles, and official Go (Golang) documentation. Methods included analysis, deduction, comparison, description, and software implementation. The result was a proprietary password manager with secure storage features, available at GitHub repository. Key advantages include compatibility with Windows and Linux, local encrypted storage for constant access, multi-sequence encryption for enhanced security, and the use of hash functions to safeguard against AES256 brute-force attacks. The study successfully achieved its goal of developing a secure password management tool.

INTRODUCTION

Today, more and more different services provide their services using the Internet (Akhmetshin et al., 2024; Kassenova et al., 2020; Kenzhin et al., 2021). To use these services, users need to authenticate using passwords – data, the knowledge of which confirms the identity of the user on the online resource (Muyang et al., 2023; Petrina et al., 2024).

To increase the security of users' personal data (full name, address, contact information, etc.), many online services require the use of passwords that have certain length (at least 10 – 12 characters), contain uppercase and lowercase letters, numbers and special characters – ~!@#$%^&*()-+{}";:.,/ (10Gaurds, 2021). This fact makes the process of remembering passwords or storing them on paper quite a difficult task.

Software tools called *password manager* are used store to securely passwords on the digital devices. Password manager is software tool that uses cryptographic methods of information protection for secure data storage and presents this data if necessary (Schneier, 2016).

Although the need to use password managers is fully expressed now, many users have justified distrust of providing their information to software tools from various companies in connection due to multiple reports of information leakage or sales.

The purpose of this investigation is software development of password manager using cryptographic algorithms of information security for data protection.

The aim of the work identified the following tasks:

- To analyze the existing software solutions used for secure password storage;

- To implement the software tool used for secure password storage.

METHODOLOGY

- The object of this work is the data provided by the user for storage.
- The subject of this work is the process of data storing and processing, which are provided by the user when password manager application.
- The information base of this work is the monographs, the scientific articles on the research topic, and the official documentation for Go (Golang) programming language.
- The methodological basis of the work includes the following scientific methods: analysis, deduction, comparison, description and software implementation.

RESULTS

The analysis of software solutions

In the modern world, where the information space is becoming overgrown with more and more data, security is becoming one of the key aspects (Cherckesova et al., 2024). In particular, the importance of passwords as the main method of ensuring information security is constantly increasing (Zhilin & Safar'yan, 2020).

The creation and use of software solutions for storing passwords is attempt to facilitate the management of these critically important elements. However, there are number of questions arise that call into the question the safety of these solutions.

One such issue is the potential control of password storage platform data by third-party organizations, including governments. Some precedents and statistics suggest that, for example, American government may interfere in various software solutions. This implies the theoretical possibility of accessing critical user data. If the risk of spyware is added, the picture becomes even more uncertain. These aspects suggest that existing software solutions for storing passwords may not be completely secure.

The conclusion is simple: the topic of security of software solutions for storing the passwords requires more in–depth research and discussion. We will highlight this issue in more detail in the following sections of the article. Examples of the most used software tools used to securely store user data are:

- Dashline – developed by Dashline;
- 1Password – 1Password company development;
- Icloud KeyChain is developed by Apple.

Dashline

Dashline is software development by Dashline Company. Three methods are supported as authentication methods: two–factor authentication, PIN–code, and biometrics. The interface of the software is shown in the Figure 1.

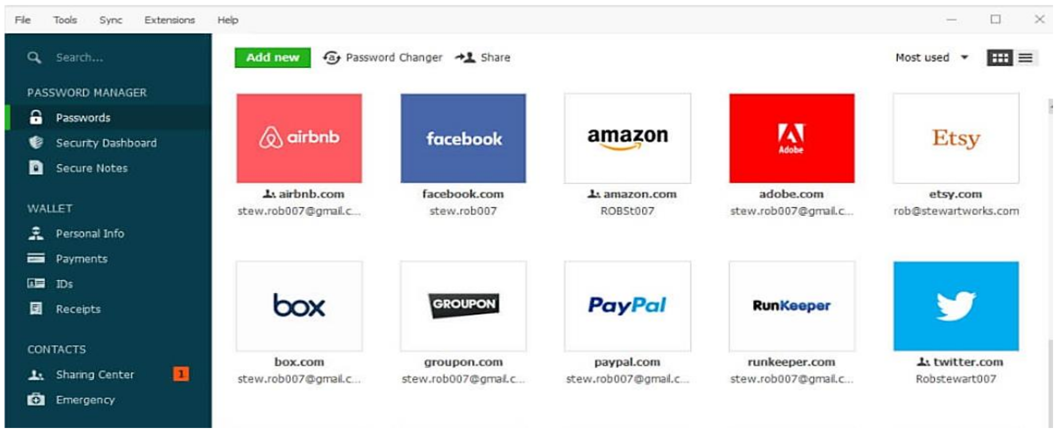


Figure 1: Dashline software interface

User data is stored with application of remote cloud storage technology. Therefore, in order to ensure secure exchange between the storage and the user, VPN module using the cryptographic methods of information protection is built into the software in question.

The advantages of the software in question are:

- Availability of the free usage plan (tariff);
- Presence of the built-in VPN module;
- Good rating of the manufacturing company.

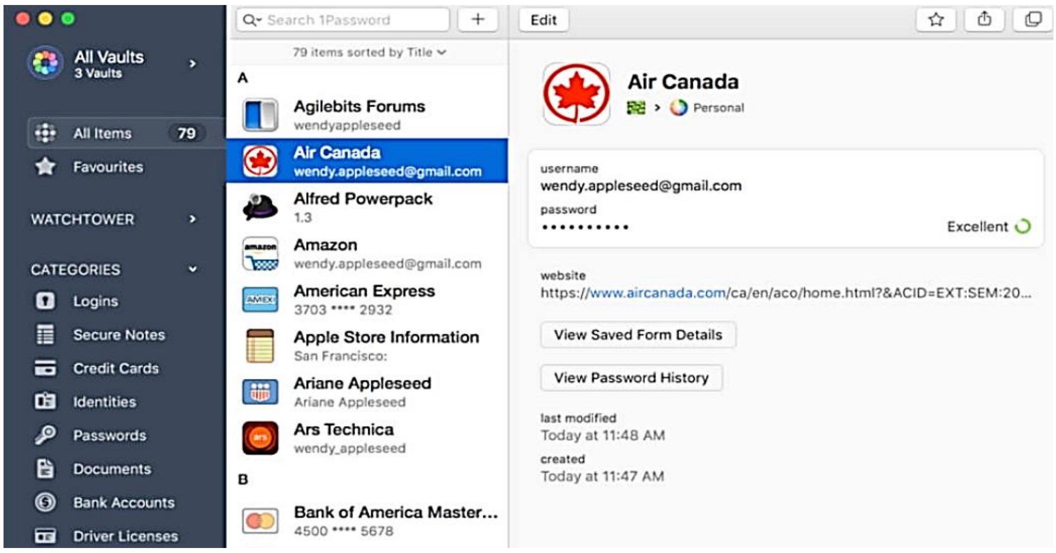
The disadvantages of the software in question are:

- Lack of Russian interface;
- Strong limitation of the free tariff functionality;
- Lack of access to the data without Internet access.

1Password

1Password is the software development by 1Password Company. The master–password of the user is applicated to the access to the saved information.

The interface of the software is demonstrated in the Figure 2.



A

Agilebits Forums
wendyappleseed

Air Canada
wendy.appleseed@gmail.com

Alfred Powerpack
1.3

Amazon
wendy.appleseed@gmail.com

American Express
3703 **** 2932

Apple Store Information
San Francisco:

Ariane Appleseed
Ariane Appleseed

Ars Technica
wendy_appleseed

B

Bank of America Master...
4500 **** 5678

Air Canada
Personal

username
wendy.appleseed@gmail.com
password
•••••••••• Excellent

website
https://www.aircanada.com/ca/en/aco/home.html?&ACID=EXT:SEM:20...

View Saved Form Details

View Password History

last modified
Today at 11:48 AM
created
Today at 11:47 AM

Figure 2: Interface of the 1Password software tool

The AES–256 cryptographic function is used to ensure the security of stored data. Unlike the Dashline software tool, the 1Password software tool allows to select the data storage location: in the cloud or in the local storage.

The advantages of the software in question are:

- Possibility of choosing the type of data storage.
- Availability of notifications about data leakage of the online service.

The disadvantages of the software in question are:

- Lack of free usage tariff;
- There is no possibility of increasing the space in the cloud.

Icloud keychain

Icloud KeyChain is software development by Apple Company. Access to the stored information is used by means of authentication of the device owner: Pin–code, master– password, biometrics (Touch_ID, Face_ID).

The interface of the software is shown in Figure 3.

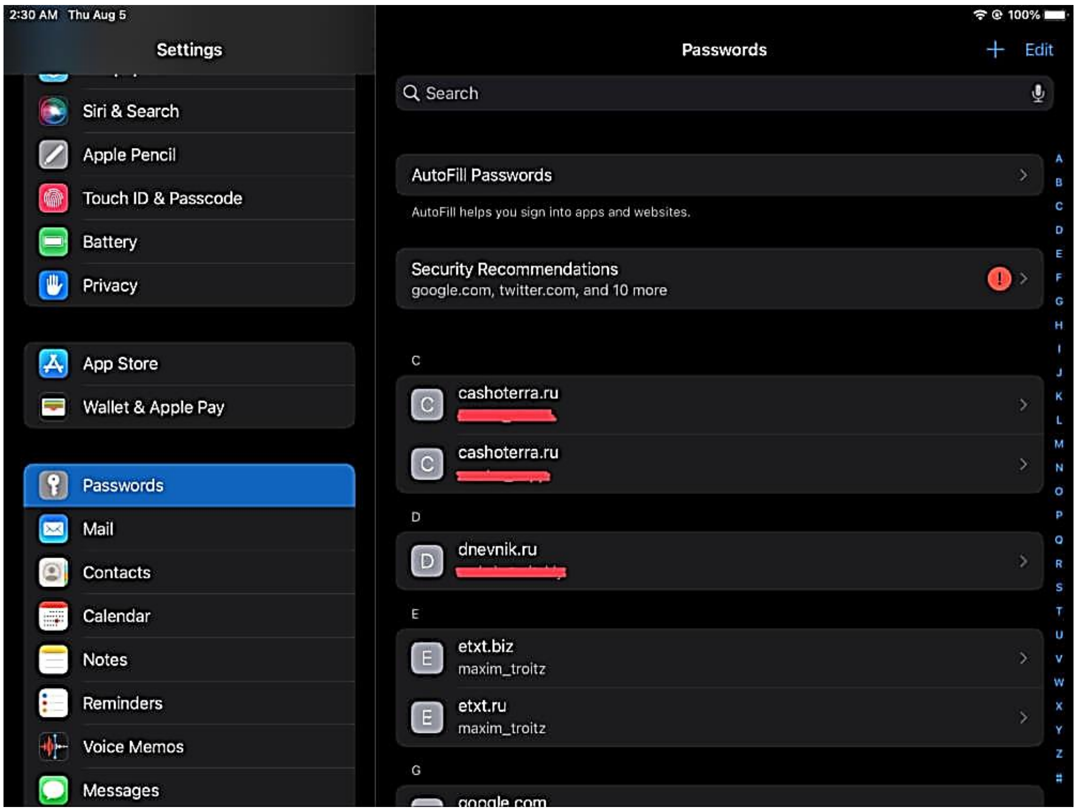


Figure 3: Icloud KeyChain software interface

Security of stored data is ensured by encrypting AES–256 cryptographic algorithm.

The advantages of the software in question are:

- Updating data between devices of the same user.
- Availability of notifications about data leakage of online services.

The disadvantages of the software in question are:

- Use is possible only on Apple devices.

Software implementation of the password manager

Main characteristics of the software implementation

After analyzing the existing solutions used for the secure storage of user data, the following characteristics of the software solution being developed were formulated:

- Ensuring the possibility of use on the various types of operating systems.
- Creation of local storage of user data with the provision of choice of storage path.
- Using the cryptographic data protection methods to ensure the security of stored data.
- Providing the ability to use different passwords to store different data.

One of the main characteristics of the software being developed is the possibility of using it on various types of operating systems, such as Windows, Linux, MacOS, etc. (Tanenbaum, 2018). To ensure the fulfillment of this property, it is necessary to select the compiled development language, the result of which is the executable (binary) file. The source code compiled into executable file, taking into account the specifics of the operating system, does not require the use of programming language interpreter.

The Go language (Golang), developed by Google in 2009, was chosen as the language for developing its own password manager (The Golang Programming Language. Official website, n.d.). At the changing the internal variables of the GOOS language (the type of operating system) and GOARCH (the architecture of the processor), it can be create the executable file for different operating systems. The chosen development language has large standard library that allows creating the reliable and readable program code.

To ensure the safe storage of user information, software development uses the cryptographic data protection methods. After analyzing the existing products that provide the functions of password managers, was selected: AES256 cryptographic encryption algorithm; GCM cryptographic mode of operation; SHA256 hash function.

The AES256 cryptographic encryption algorithm is the symmetric encryption algorithm adopted as standard in 2000 (Omasson, 2022). The symmetry property of this encryption algorithm indicates that the single encryption key is used during encryption and decryption of data. The use of symmetric algorithm in the software implementation of the password manager is due to the speed and reliability of this algorithm. Since, when encrypting the identical data using the same encryption key, the result is identical ciphertext, the GCM operating mode (regime) is used to ensure greater security of the user's stored data.

GCM (Galois Counter Mode) is operation mode (regime) of encryption algorithms, in which the random sequence of data is added, which changes the general appearance of encrypted data (Omasson, 2022). To generate the encryption key, the sequence entered by the user when working with the software implementation is applicated.

Custom sequence is used as argument to the SHA256 hash function. Hash function is the function whose argument is data of any length, and as result it generates the special sequence of bytes of certain length (the result of the SHA256 hash function is sequence of 32 bytes) (Omasson, 2022). The properties that make the SHA256 hash function usable as encryption key generator of the AES256 cryptographic algorithm are: 1) Minimal change in the hash function argument (changing one bit of data) leads to the strong change in the result. 2) It is not possible to get the value of hash function argument by analyzing the resulting byte sequence. 3) It is impossible to match the value of hash function argument to obtain the known byte sequence.

To use these methods to ensure the security of user data, the standard library of Go programming language contains the following packages: Crypto/AES – the software implementation of the AES256 cryptographic algorithm, Crypto/Cipher – the software implementation of the GCM operating mode and, and Crypto/SHA256 – software implementation of the SHA256 hash function. Command line interface has been implemented to interact with the software development for various types of operating systems. The scheme of password manager using is shown in Figure 4:

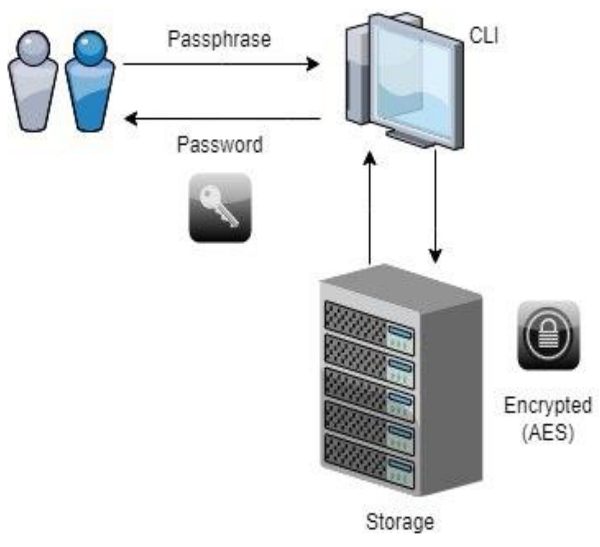


Figure 4: The scheme of using the developed password manager

Software implementation of the password manager

Algorithm of the software implementation of password manager is shown in Figure 5:

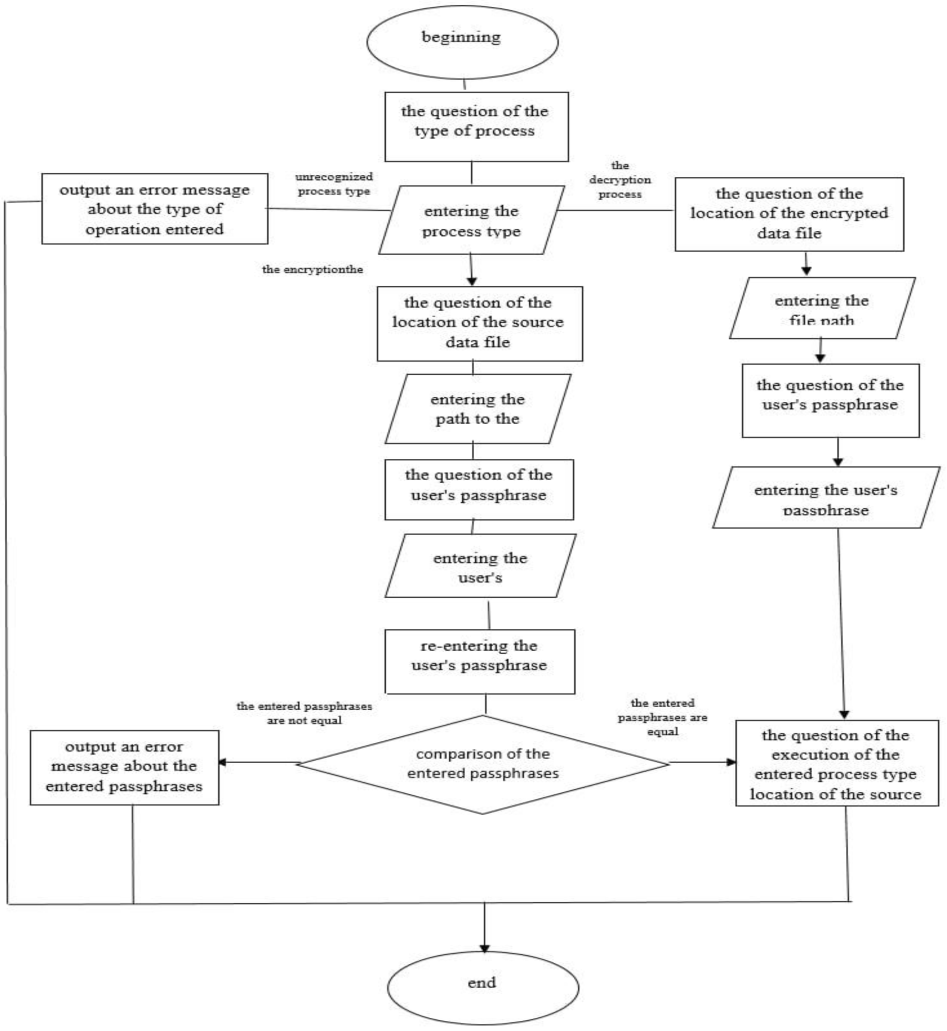

beginning
the question of the type of process
output an error message about the type of operation entered
unrecognized process type
entering the process type
the decryption process
the question of the location of the encrypted data file
the encryptionthe
the question of the location of the source data file
entering the file path
entering the path to the
the question of the user's passphrase
the question of the user's passphrase
entering the user's passphrase
entering the user's
re-entering the user's passphrase
the entered passphrases are not equal
the entered passphrases are equal
output an error message about the entered passphrases
comparison of the entered passphrases
the question of the execution of the entered process type location of the source
end

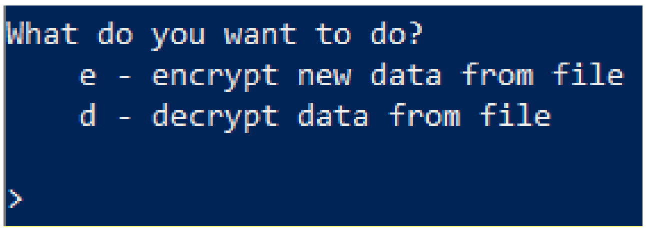
Figure 5: The scheme of the implemented password manager algorithm

During the software development of the password manager, the following modules were developed: options, encryption and decryption.

The options module implements the following tasks:

- Getting the type of process.
- Getting the path of the file containing the necessary information.
- Getting the user's passphrase.

The process type is obtained using the query shown in Figure 6:

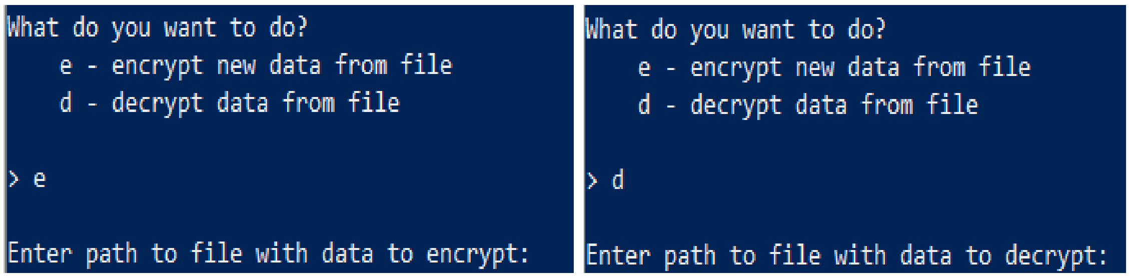


```
What do you want to do?
    e - encrypt new data from file
    d - decrypt data from file

>
```

Figure 6: Message for entering the type of process

As it can be seen from the figure above, to start the data encryption process user must enter letter 'e'. To start the decryption process, enter the letter ’d’. If the process type is entered incorrectly, user receives the message "unidentified option request".

After selecting the process type, the message is displayed requesting the full path to the file containing the data, which are necessary for the operation of selected process. If there is error reading data from the file, the user receives the message "error occurred while reading data from the full path of the file". The queries are shown in Figure 7:



```
What do you want to do?
    e - encrypt new data from file
    d - decrypt data from file

> e

Enter path to file with data to encrypt:
```

```
What do you want to do?
    e - encrypt new data from file
    d - decrypt data from file

> d

Enter path to file with data to decrypt:
```

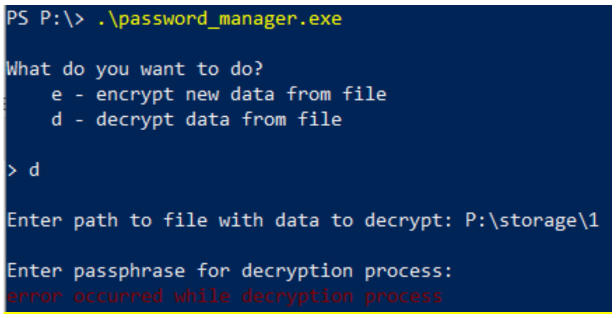
Figure 7: Requesting of the full paths after selecting the process type

When entering the user passphrase, the phrase itself is not displayed in the terminal being used, to ensure that this phrase is safe from intruders. The module used to enter securely the passphrase has the name golang.org/x/term.

The phrase is entered twice during the data encryption process. Re-entering the passphrase is necessary to confirm the correctness of the phrase used, because the data encrypted with this phrase cannot be restored without the user's knowledge of it.

During the data decryption, if the data decryption error occurred when using the hash value of the user passphrase as the encryption key (incorrect passphrase was used during data decryption), the user will receive the message "error occurred while decryption process", without receiving any information about the encrypted data.

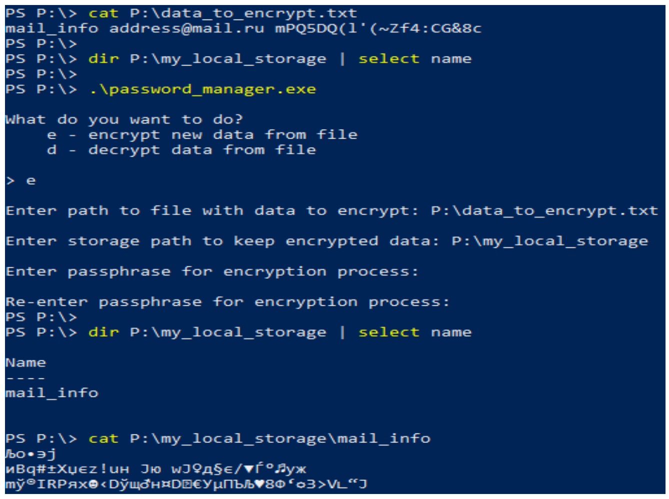
This behavior is shown in Figure 8:



```
PS P:\> .\password_manager.exe

What do you want to do?
    e - encrypt new data from file
    d - decrypt data from file

> d

Enter path to file with data to decrypt: P:\storage\1

Enter passphrase for decryption process:
error occurred while decryption process
```

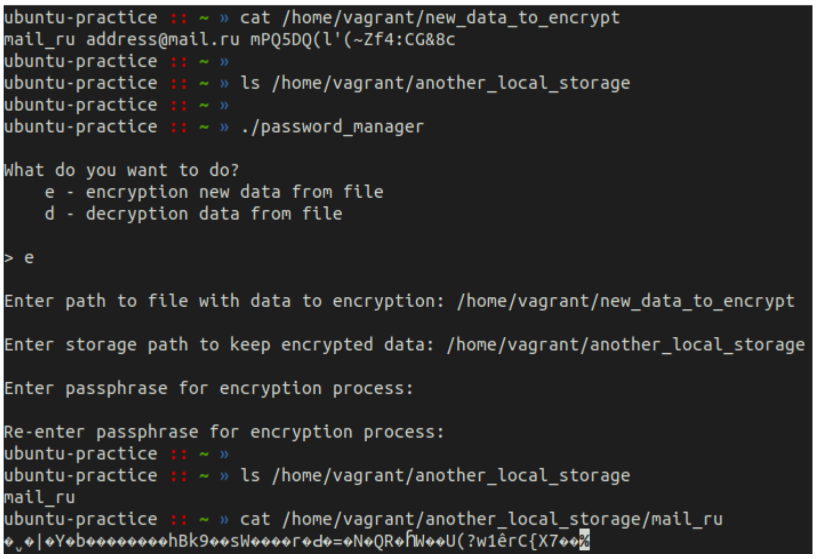
Figure 8: Data decryption error

Thus, this example demonstrates that different passphrases can be used for different files, so that when one of them is exposed, no encrypted data can be decrypted. Figures 9 and 10 demonstrates the examples of using the software development to encrypt data on Windows and Linux operating systems.



```
PS P:\> cat P:\data_to_encrypt.txt
mail_info address@mail.ru mPQ5DQ(l'(~Zf4:CG&8c
PS P:\>
PS P:\> dir P:\my_local_storage | select name
PS P:\>
PS P:\> .\password_manager.exe

What do you want to do?
    e - encrypt new data from file
    d - decrypt data from file

> e

Enter path to file with data to encrypt: P:\data_to_encrypt.txt

Enter storage path to keep encrypted data: P:\my_local_storage

Enter passphrase for encryption process:

Re-enter passphrase for encryption process:
PS P:\>
PS P:\> dir P:\my_local_storage | select name

Name
----
mail_info


PS P:\> cat P:\my_local_storage\mail_info
љo•эj
иBq#±Xџєz!uн Јю wЈ♀д§є/▼ѓ°♬уж
mў®IRPяx☻‹DўщđнxD⊡€УµПЊЉ♥8Ф'¤З>VL“Ј
```

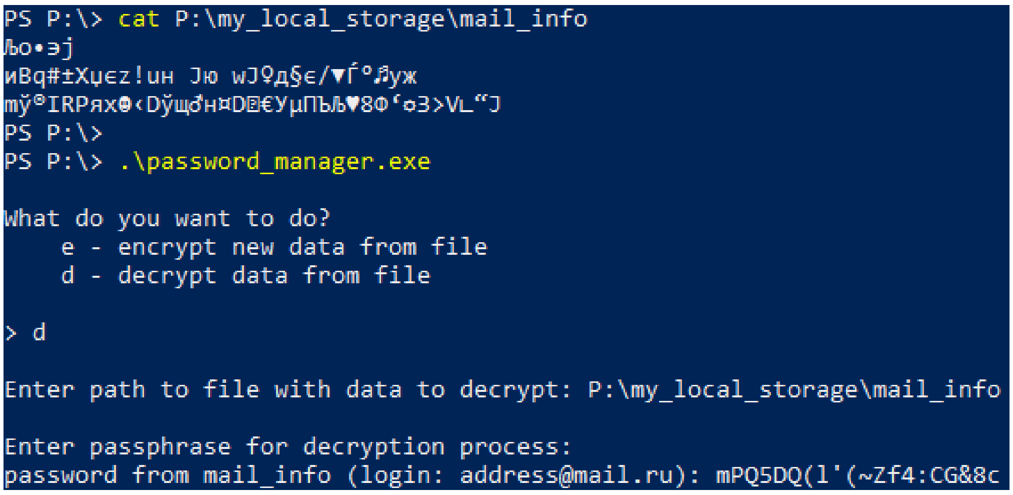
Figure 9: Example of data encryption on Windows OS



```
ubuntu-practice :: ~ » cat /home/vagrant/new_data_to_encrypt
mail_ru address@mail.ru mPQ5DQ(l'(~Zf4:CG&8c
ubuntu-practice :: ~ »
ubuntu-practice :: ~ » ls /home/vagrant/another_local_storage
ubuntu-practice :: ~ »
ubuntu-practice :: ~ » ./password_manager

What do you want to do?
    e - encryption new data from file
    d - decryption data from file

> e

Enter path to file with data to encryption: /home/vagrant/new_data_to_encrypt

Enter storage path to keep encrypted data: /home/vagrant/another_local_storage

Enter passphrase for encryption process:

Re-enter passphrase for encryption process:
ubuntu-practice :: ~ »
ubuntu-practice :: ~ » ls /home/vagrant/another_local_storage
mail_ru
ubuntu-practice :: ~ » cat /home/vagrant/another_local_storage/mail_ru
�ˬ�|�Y�b���������hBk9��sW�����r�d�=�N�QR�ꟻW��U(?w1êrC{X7��%
```

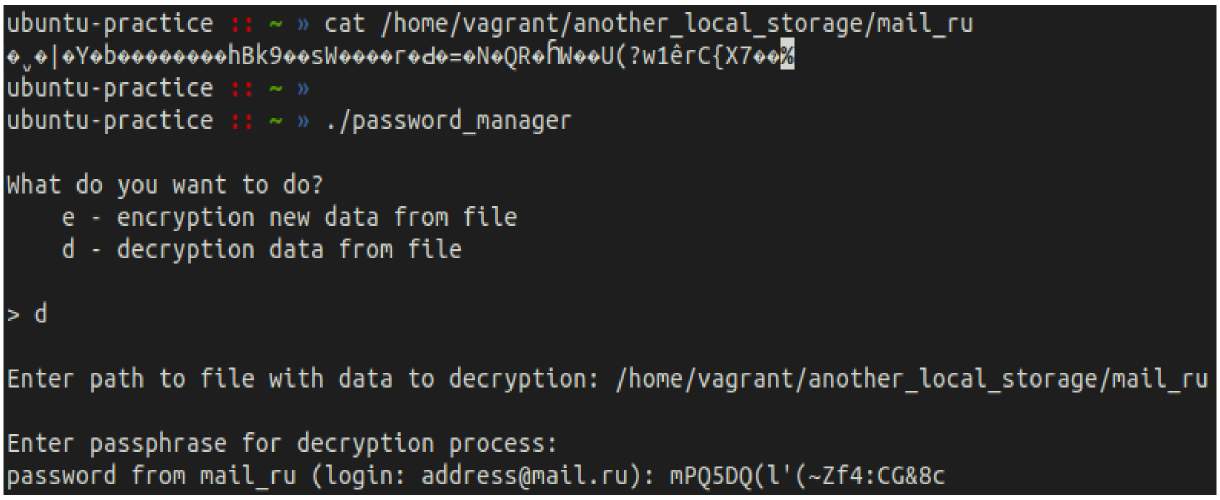
Figure 10: Example of data encryption on OC Linux

As can be seen from the examples shown above, the result of data encryption is the file with the name specified in the source data file (mail_ru) and containing the encrypted specified username and password. Figures 11 and 12 show the examples of using software development to decrypt data contained in the previously created files on Windows and Linux operating systems.



```
PS P:\> cat P:\my_local_storage\mail_info
Љo•эj
иBq#±Xџєz!uн Јю wЈ♀д§є/▼ѓ°♫yж
mў®IRPяx☻‹DЎщђн¤D⊡€УµПЊЉ♥8Ф‘¤З>VL“Ј
PS P:\>
PS P:\> .\password_manager.exe

What do you want to do?
    e - encrypt new data from file
    d - decrypt data from file

> d

Enter path to file with data to decrypt: P:\my_local_storage\mail_info

Enter passphrase for decryption process:
password from mail_info (login: address@mail.ru): mPQ5DQ(l'(~Zf4:CG&8c
```

Figure 11: Example of decrypting data on Windows OS



```
ubuntu-practice :: ~ » cat /home/vagrant/another_local_storage/mail_ru
�ˬ�|�Y�b���������hBk9��sW�����r�d�=�N�QR�ꟻW��U(?w1êrC{X7��%
ubuntu-practice :: ~ »
ubuntu-practice :: ~ » ./password_manager

What do you want to do?
    e - encryption new data from file
    d - decryption data from file

> d

Enter path to file with data to decryption: /home/vagrant/another_local_storage/mail_ru

Enter passphrase for decryption process:
password from mail_ru (login: address@mail.ru): mPQ5DQ(l'(~Zf4:CG&8c
```

Figure 12: Example of decrypting data on OC Linux

As can be seen from the examples shown above, the result of decrypting data using the developed software password manager is the message in the format "password from file_name (login: decryption_login): decryption_parol".

Output by section

The result of the software development is the software tool that can be used in various operating systems (Windows and Linux). This software tool uses cryptographic data protection methods to store securely the encrypted data.

CONCLUSION

In the result of the work done, the existing programming tools used for secure password storage were analyzed. The proprietary password manager, which ensures the secure storage of information, has been implemented and described. The source code is presented on the online repository https://github.com/zexy-swami/password_manager.

The advantages of this software implementation are:

1. Possibility of using the password manager developed by the authors on the various types of operating systems – Windows and Linux.
2. Property of local storage of encrypted information, providing constant access to it.
3. The ability to use the different sequences instead of single master key, which are used to encrypt and decrypt the user's data.
4. Using of hash function when creating the encryption / decryption key to increase the security of encrypted data against the attack of the full iteration of the key values of the AES256 cryptographic algorithm.

Thus, the purpose of the study was achieved.

REFERENCES

10Gaurds. Kak sozdat' nadezhnyy parol': Podrobnoye rukovodstvo [How to create strong password: Detailed guide]. June 21, 2021. https://10guards.com/ru/articles/how-to-create-a-strong-password-step-by-step-guide/ (accessed on March 20, 2022).

Akhmetshin E, Nemtsev A, Shichiyakh R, Shakhov D, Dedkova I. Evolutionary algorithm with deep learning based fall detection on Internet of Things environment. Fusion Pract Appl 2024,14:132-45. http://dx.doi.org/10.54216/FPA.140211

Cherckesova L, Revyakina E, Safaryan O, Porksheyan V, Kazaryan M. Analysis of the possibilities of carrying out attacks on the functions of transferring control to operating system console using active intelligence methods. Int Res J Multidisc Scope 2024,5:516-34. https://doi.org/10.47857/irjms.2024.v05i02.0558

Kassenova G, Zhamiyeva A, Zhildikbayeva A, Doszhan R, Sadvakassova K. Digitalization of the company's financial resources (by the example of Air Astana JSC). E3S Web Conf 2020,159:04021. http://dx.doi.org/10.1051/e3sconf/202015904021

Kenzhin ZB, Tulegenova AU, Zolkin AL, Kosnikova OV, Shichkin IA. Labour market under economy digitalization. E3S Web Conf 2021,311:8007. http://dx.doi.org/10.1051/e3sconf/202131108007

Muyang G, Sekerin V, Efremov A, Gorokhova A, Gayduk V. Legal basis for the development of an industrial internet platform in the context of digital transformation. Rev Jurid 2023,3:667-78.

Omasson J. O Kriptografii Vser'yez [About Cryptography Seriously]. Moscow: DMK Press; 2022. 328 p.

Petrina O, Stadolin M, Kozhina V, Kurtynov I, Nikolskaya E, Orlova E. Bank financial risk assessment in the digital background. Int J Saf Secur Eng 2024,14:765-71. https://doi.org/10.18280/ijsse.140309

Schneier B. Prikladnaya Kriptografiya. Protokoly, Algoritmy, Iskhodnyye Teksty na Yazyke Si [Applied Cryptography. Protocols, Algorithms and Source Code in C]. Moscow: Dialectics; 2016. 1024 p.

Tanenbaum E. Sovremennyye Operatsionnyye Sistemy [Modern Operating Systems]. Peter: Classics of Computer Science; 2018. 920 p.

The Golang Programming Language. Official website. n.d. https://go.dev (accessed on March 21, 2022).

Zhilin V, Safar'yan O. Artificial intelligence in data storage systems. Vestnik of Don State Technical University 2020,20(2):196–200. https://doi.org/10.23947/1992-5980-2020-20-2-196-200