Design and Development of a Secure Password Manager Using End-to-End Encryption

Prakash Mali[1], Sanskar Patil[2], Prathmesh Kale[3], Rohnit Setia[4]
[1,2,3,4]*JSPM's RSCOE Polytechnic*

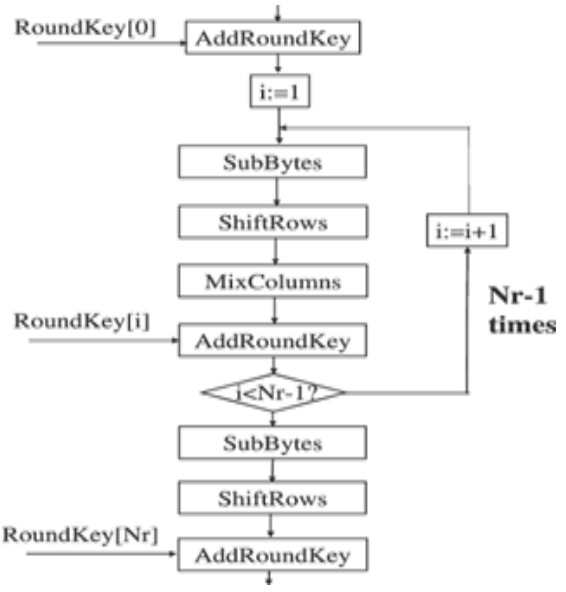
***Abstract*—This paper describes the design and implementation of a secure password manager application that uses end-to-end encryption to protect users' credentials. As the number of users for online services increases, they need to remember various usernames and passwords. For this reason, most people use some insecure practices in their daily lives, such as using the same password for different accounts or storing them in plaintext. In this proposed system, all the sensitive information is encrypted at the client side with AES-256 encryption generated from the master password. This means that passwords are never persisted in readable format. Even if an unauthorized party obtains access to the stored data, it will remain unreadable without the correct master password. It includes a basic, responsive interface that allows users to add, view, and delete stored credentials securely. Due to its security, privacy, and usability, this system is suitable for personal credential management.**

***Index Terms*—Password Manager, End-to-End Encryption, Cyber Security, Data Privacy, AES Encryption**

I. INTRODUCTION

Nowadays, people use numerous online facilities like social networks, banking services, e-mails, and online shopping websites. Every facility needs log-on information-a great problem to keep track of passwords. Unluckily, most users are into unsafe practices, such as password reuse or writing them down in unsecured locations, thus being highly vulnerable to cyber-attacks.

A password manager securely keeps login credentials in one place. However, most solutions today rely on cloud or server-side storage, which comes with a host of concerns regarding data privacy and unauthorized access. This project targets the implementation of a client-side password manager that ensures end-to-end encryption, wherein sensitive data can be encrypted before being stored and decrypted only by an authorized user.



Flow chart

II. PROCEDURE FOR PAPER SUBMISSION

A. Review Stage

Submit your manuscript electronically for review. prepare it in two-column format, including figures and tables(untill it don't fit properly and data is not visible).

B. Final Stage

After your paper has been accepted. The authors of the accepted manuscripts will be given a copyright form and the form should accompany your final submission.

C. Figures

As said, to insert images in Word, position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard

and then Edit | Paste Special | Picture (with Float over text‖ unchecked).

III. MATH

Although the proposed password management system is primarily software-based, mathematical concepts are involved in the encryption and key generation process.

Let:

- P be the master password entered by the user
- K be the cryptographic key derived from the master password
- D be the plaintext credential data
- E(D, K) be the encryption function
- C be the encrypted data

The encryption process can be represented as:

C=E(D,K)

The decryption process is represented as:

D=D(C,K)

Where AES-256 encryption is used for the function E. Without the correct key K, it is computationally infeasible to retrieve the original data D, ensuring confidentiality and data security.

IV. UNITS

The system uses the following units and measures:

Bits (bit): They are used to represent encryption strength. E.g. AES-256

Bytes: The unit used to measure encrypted data stored.

Milliseconds (ms): resolution used to represent encryption and decryption time

Characters: Refers to password length

These are standard units that ensure consistency in security and performance evaluation.

- Units Used in the System

	Unit	Description
1	Bit	Encryption strength
2	Byte	Data size
3	Ms	Time measurement
4	character	Password length

V. HELPFUL HINTS

The following tips have been considered during the design and implementation of the secure password management system. This might be helpful for future developers and users in improving security and usability.

The user should create a strong master password, including a mixture of uppercase letters, lowercase letters, numbers, and special characters to enhance the strength of encryption.

Also, a master password should never be stored as plain text and shared with anyone.

All the encryption and decryption processes should be done directly on the client side without any interference, for the fact of ensuring end-to-end encryption.

Deleting unused or invalid credentials regularly can reduce security risks.

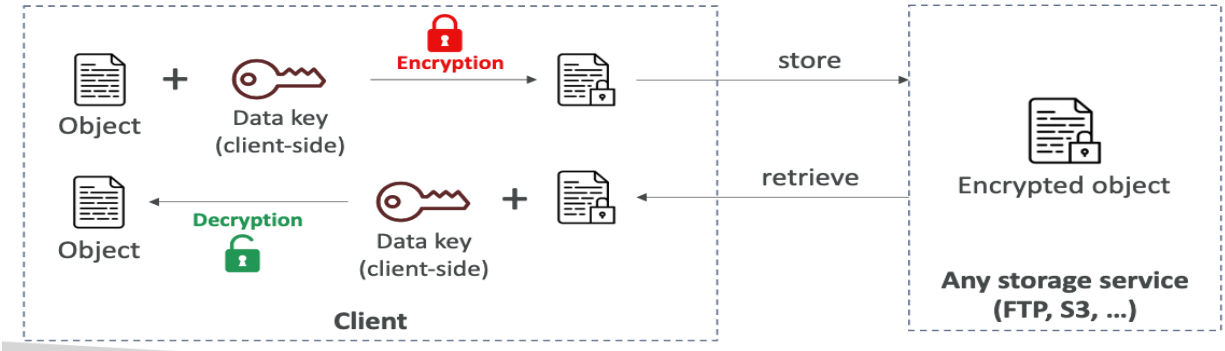
Validation of the input fields should be performed to prevent incorrect entry, including harmful data entry.

Application interface should remain simple and intuitive to reduce the amount of user mistakes.

Users must avoid accessing the password manager on public or shared devices.

Any additional backup mechanisms must be set up using encrypted storage.

These tips ensure data confidentiality, integrity, and overall system security.



VI. PUBLICATIONPRINCIPLES

The research presented here is bound by the standard principles of publication, hence ensuring academic honesty, originality, and clarity. The content is original, without any trace of plagiarism, and has been prepared as such. The technical concepts are described transparently and coherently for better readability and understanding.

The paper strictly adheres to the required journal format, with the paper organized into well-defined sections: introduction, proposed system, methodology, results, and conclusion. Ethical considerations about data privacy and security have been followed in every step of this study. No real user data was gathered or exposed during system implementation.

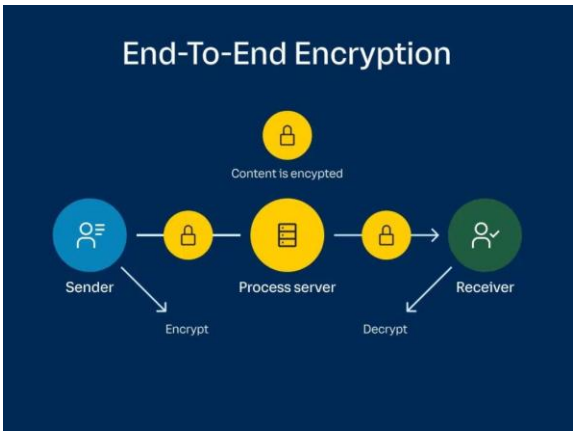
Proper formatting guidelines, including font style, figure numbering, table labeling, and reference citation, have been followed according to journal requirements. This work aims to contribute meaningfully to the realms of cyber security and password management systems.

VII. CONCLUSION

This paper presented the design and implementation of a secure password management system using end-to-end encryption. The proposed system addresses common security issues linked to traditional password storage methods by encrypting all sensitive credentials on the client side. Using a master password-based encryption mechanism, it ensures that passwords stored cannot be accessed by unauthorized users.

The use of AES-256 encryption ensures strong security while keeping performance efficient. The system avoids storing data on the server side, reducing the possibility of data breaches and hence improving the privacy of the users. The experimental results show that even if unauthorized access is given to the storage, the encrypted data remains secure.

Overall, the developed password manager provides a robust, easy-to-use, and secure means of managing credentials. It shows the importance of encryption-based security in contemporary applications and can provide a very good foundation for further extension, such as further support of biometric authentication based on cloud synchronization with encryption of data and multi-factor security mechanisms.



VIII. APPENDIX

This appendix summarizes key implementation and security aspects of the secure password management system. The application makes use of only standard web technologies (HTML, CSS, JavaScript), which ensures that it is platform-independent and easy to use.

All stored credentials are protected through AES-256 encryption in the system. The user master password is used to derive an encryption key that is not stored in the application. Encryption and decryption both happen on the client side to allow end-to-end encryption.

The credentials are encrypted and stored locally. Access to passwords and even system settings is strictly guarded through authentication. The user interface is simplistic and mobile-friendly, allowing users with minimal interaction to securely add and view or delete credentials. These implementation choices provide improvement in the field of data security, protection of user privacy, and reliability of system performance.

ACKNOWLEDGMENT

The author is grateful to the project guide for his immense guidance, constant support, and valuable suggestions which helped throughout the project development. Thanks are also due to the faculty members of the department for their encouragement and technical support. The author would also want to acknowledge the institution for offering relevant facilities and an environment that enabled me to complete this work successfully. Last but not least, I would like to thank friends and family members who motivated and supported me during this project.

REFERENCES

[1] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed., Pearson Education, 2017.

[2] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 2018.

[3] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," FIPS PUB 197, 2001.

[4] OWASP Foundation, "OWASP Top Ten Web Application Security Risks," 2023. [Online]. Available: https://owasp.org

[5] Mozilla Developer Network (MDN), "Web Cryptography API," [Online]. Available: https://developer.mozilla.org

[6] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, Wiley, 2016.

[7] ISO/IEC 27001, "Information Security Management Systems," International Organization for Standardization, 2022.

[8] A. Green and S. Smith, "Client-Side Encryption for Secure Web Applications," International Journal of Computer Security, vol. 10, no. 3, pp. 45–52, 2021.