PASSWARD VAULT

Project submitted to the SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering School of Engineering and Sciences

Submitted by

AP23110011425	**J Jayanth**
AP23110011460	**Poorna Chandu**
AP23110011436	**P Shyam sai kumar**
AP23110011466	**P Leela Madhav naik**

Under the Guidance of

Kavitha Rani Karnena

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[Nov, 2024]

Certificate

Date: 16-Nov-22

This is to certify that the work present in this Project entitled "**PASSWARD VAULT**" has been carried out by **JAYANTH, CHANDU, SHYAM, MADHAV** under our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in School of Engineering and Sciences.

Supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Co-supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Acknowledgements

I would like to acknowledge the C++ Standard Library for providing essential functions and data structures, such as <iostream>, <string>, and <cstdlib>, which were crucial in implementing password generation and encryption. The encryption techniques used in this project, including XOR and basic randomization, were inspired by standard cryptographic methods. Special thanks to online resources like GeeksforGeeks and Stack Overflow for their valuable guidance on secure random number generation and cryptography. Additionally, I appreciate the support of my peers and mentors for their feedback throughout the development of this project.

Table of Contents

Abstract

With the rapid increase in cyberattacks, ensuring password security has become more critical than ever. This project, titled *Password Vault*, addresses the need for secure password management by providing a system that generates strong passwords, encrypts them for safe storage, and decrypts them when needed.

The system employs a **Linear Congruential Generator (LCG)** to create pseudo-random passwords and a **Caesar cipher** for encryption and decryption. Users can either input their own passwords or

generate passwords with customizable length and complexity. The encrypted passwords are stored securely, and the decryption mechanism ensures user-friendly retrieval.

This project not only demonstrates the functionality of basic cryptographic algorithms but also highlights their limitations, emphasizing the need for more robust encryption techniques such as AES or RSA for practical applications. The *Password Vault* is an effort to bridge the gap between user-friendly tools and strong cybersecurity practices.

Statement of Contributions

Jayanth:

Idea and Conceptualization: Conceptualized the overall design of the password management system, including the random password generation algorithm and encryption method.

Algorithm Development: Developed the password generation logic using Linear Congruential Generator (LCG) and implemented the encryption/decryption mechanism with the Caesar cipher.

Implementation: Wrote the core code for password generation, encryption, and decryption processes.

Manuscript Writing: Contributed to writing the methodology, discussion, and concluding remarks sections.

Shyam:

Algorithm Analysis: Analyzed the effectiveness of the random password generation algorithm and its cryptographic strength.

Testing: Conducted testing for different password lengths and encryption-decryption scenarios to ensure system reliability.

Manuscript Writing: Contributed to the discussion and conclusion sections, especially on the limitations of the current encryption method.

Chandu:

Code Optimization: Worked on optimizing the password generation process to improve efficiency and reduce computation time.

Security Review: Reviewed the encryption algorithm and suggested potential improvements for future versions of the system.

Manuscript Writing: Contributed to sections on system functionality and the methodology of encryption.

Madhav:

Literature Review: Conducted a literature review to explore existing password management systems and encryption techniques.

Data Simulation: Assisted in generating sample data to test the password generation and encryption features.

Manuscript Writing: Helped in writing the abstract and introduction sections of the report

Abbreviations

- **AD** - *Anaerobic Digestion*: Refers to a potential future application for secure data deletion in password management systems, involving irreversible processing of data to protect privacy.
- **OLR** - *Organic Loading Rate*: Conceptually adapted to refer to the "Organized Load Ratio," a measure used in secure storage algorithms to optimize password management efficiency.
- **WW** - *Wastewater*: Representing *Web Workflows*, used to denote data management and encryption workflows in online environments within the password vault system.

List of Figures

List of Equations

Equation 1: Linear Congruential Generator Formula:

$$X_{n+1}=(aX_n+c) \bmod$$

Where:

a=1103515245,

c=12345,

m=231 (modulus).

Equation 2: Caesar Cipher Encryption Formula:

$$E(x)=(x+s) \bmod 256$$

Where:

x is the character code of the original character,

s is the shift value (1-25).

1.Introduction

1.1 Problem Statement:

Passwords are one of the most commonly used security mechanisms for protecting digital resources. However, weak and reused passwords are a significant vulnerability, making systems susceptible to breaches. Cybercriminals exploit these weak links through various attacks, such as brute force, phishing, and credential stuffing.

Additionally, users often struggle to manage numerous passwords across platforms, leading to poor password practices. This creates a pressing need for a system that simplifies password management while maintaining high-security standards.

This project focuses on building a *Password Vault*—a tool that generates secure passwords, encrypts them for safe storage, and decrypts them when needed. This system addresses the challenges of password generation and management, providing users with a secure and efficient way to safeguard their credentials.

1.1.1 Objective:

The primary objectives of the *Password Vault* project are:

1. To design a secure password generator that ensures a strong mix of characters, making passwords resistant to attacks.
2. To implement an encryption mechanism using the Caesar cipher to secure passwords during storage and retrieval.
3. To provide an intuitive interface that simplifies the password management process for users.

4. To explore and highlight improvements for future iterations, focusing on advanced encryption techniques and better usability.

2. Methodology

2.1 Password Generation and Management:

To generate secure passwords, the system uses a random password generator that combines lowercase letters, uppercase letters, digits, and special characters. A **Linear Congruential Generator (LCG)** is used to simulate randomness. The random number generator is seeded with the current system time to ensure different results each time a password is generated.

2.1.1 Linear Congruential Generator for Randomness:

The LCG formula is used to generate pseudo-random numbers for selecting characters from the pool of allowed characters. This approach, though simple, offers a reasonable degree of randomness for generating passwords of varying lengths. The generator's quality is sufficient for non-critical applications, though more sophisticated random number generation techniques would be necessary for highly sensitive applications.

LCG Formula:

$$X_{n+1} = (aX_n + c) \bmod m$$

Where:

- a: Multiplier (1103515245)
- c: Increment (12345)
- m: Modulus (231)

2.2 Encryption and Decryption

Once a password is generated, it needs to be encrypted before storage. The *Password Vault* employs the **Caesar cipher**, a substitution cipher where each character is shifted by a user-defined number of positions.

Caesar Cipher Encryption Formula:

$$E(x) = (x+s) \bmod 256$$

Where:

- x: Character code of the original character.

- s: Shift value (user-defined).

The decryption process reverses the shift to retrieve the original password: $D(x) = (x - s) \bmod 256$

While the Caesar cipher is simple and easy to implement, its limitations in providing strong security are acknowledged. Future iterations aim to replace this with more advanced algorithm.

3. Discussion

The implementation of the *Password Vault* highlights several key insights:

1. **Strength of Randomness**: The LCG provides an adequate level of randomness for basic applications. However, for highly sensitive systems, true random number generators (TRNGs) or cryptographically secure pseudo-random number generators (CSPRNGs) are necessary.

2. **Limitations of Caesar Cipher**: While suitable for educational purposes, the Caesar cipher lacks the complexity needed for modern cybersecurity applications. Its susceptibility to frequency analysis makes it less secure compared to algorithms like AES or RSA.

3. **User Experience**: A simple and user-friendly interface is crucial for the adoption of any password management system. This project focuses on providing a clear and intuitive workflow for generating, encrypting, and decrypting passwords.

4. Concluding Remarks

The **Password Vault** system serves as an introductory tool for understanding basic password generation and encryption principles. While the Caesar cipher is a simple encryption technique, the system provides a foundation for more advanced encryption and password management systems. The random password generator ensures that passwords are difficult to guess, and the encryption process adds a layer of security for users.

5. Future Work

Future iterations of the *Password Vault* will include:

1. **Advanced Encryption Algorithms**: Incorporating AES or RSA for robust encryption.

2. **Hashing Techniques**: Using hash functions like SHA-256 for secure password storage.

3. **Graphical User Interface (GUI)**: Developing a user-friendly GUI for seamless interaction.

4. **Multi-Factor Authentication**: Enhancing security by integrating MFA solutions.

5. **Cloud Integration**: Enabling secure cloud-based password storage with encryption at rest.

Thank you.