**Personal Diary Management System with File Handling and Data Encryption for Privacy**

**Abstract-** The Personal Diary Management System is a software application built in C, aimed at providing a secure and efficient platform for managing personal diary entries. Users can easily add, edit, view, and delete their entries, while ensuring that sensitive data remains confidential. The system uses file handling to store entries in individual user files, ensuring persistence across sessions. To enhance privacy, all diary content is encrypted, making it accessible only to authorized users. The project integrates fundamental programming techniques such as file manipulation, string handling, and encryption, with a focus on user authentication to safeguard data. This system offers a simple yet effective solution for individuals who wish to maintain a private, encrypted digital diary, providing both convenience and security.

1. **INTRODUCTION**

In today’s digital age, managing personal information securely and privately is of utmost importance. A personal diary serves as a confidential space for individuals to express thoughts, emotions, and ideas. However, traditional paper-based diaries are prone to loss, theft, and damage.

JAYED AL NAHIAN

ID: 2422422642

Project Group: 08

Course: CSE115

Section: 07

Faculty: Dr. Mohammad Shifat-E-Rabbi [MSRb]

Department of Electrical and Computer Engineering

North South University

SAYEED MANIR

ID: 2421700042

Project Group: 08

Course: CSE115

Section: 07

Faculty: Dr. Mohammad Shifat-E-Rabbi [MSRb]

Department of Electrical and Computer Engineering

North South University

MD TAUSIF UDDIN AISHWARJA

ID: 2423815042

Project Group: 08

Course: CSE115

Section: 07

Faculty: Dr. Mohammad Shifat-E-Rabbi [MSRb]

Department of Electrical and Computer Engineering

North South University

PRINCE MAHMUD PAPEL

ID: 2421805042

Project Group: 08

Course: CSE115

Section: 07

Faculty: Dr. Mohammad Shifat-E-Rabbi [MSRb]

Department of Electrical and Computer Engineering

North South University

With the advent of technology, digital diaries have gained popularity, offering a more secure and easily accessible way to maintain personal records.

The Personal Diary Management System is designed to provide users with a secure and efficient platform to manage their diary entries. This system allows users to add, edit, view, and delete entries, all while ensuring the privacy of their data. Using file handling for persistent data storage and encryption techniques for safeguarding the contents, the system ensures that only authorized users can access their private entries. Moreover, by providing an easy-to-use interface, it aims to offer a simple yet effective solution to maintaining a digital diary.

The system also integrates user authentication features, where users must register and log in before they can access their diaries. This ensures that each user’s data remains private and isolated from others. With its straightforward functionality and secure data management practices, the Personal Diary Management System seeks to provide a reliable solution for individuals who wish to keep their personal thoughts and experiences safe in the digital world.

* **Command-Line Interface (CLI)**: The system is accessed via a command-line interface, providing a straightforward and effective means for user interaction. Users can easily navigate through the program using simple text-based commands. Actions such as signing up, logging in, adding new entries, and editing or deleting old ones are all managed via this interface. The minimalistic design ensures that the system is both easy to use and functional, focusing on usability and clarity.

The system is composed of various interconnected modules that ensure each component functions independently but works together seamlessly. This modularity ensures that the system remains secure, stable, and adaptable for future improvements.

**3. IMPLEMENTATION**

The Personal Diary Management System is built using the C programming language, with key features revolving around user authentication, diary entry management, and file handling. It uses basic encryption to ensure privacy and implements an intuitive user interface for interaction. The following outlines the critical components involved in the system's implementation.

* **User Authentication**

A secure login system is essential to ensure that only registered users can access their diary. The first step of the program allows a user to register an account by creating a unique username and password. The system checks if the chosen username already exists. If it does, the user is prompted to select a different one. Once the username is validated, both the username and the password are saved.

* + **Registration:** During registration, the user enters a username and password. The program ensures that the username is not already in use and stores the new credentials for future access.
  + **Sign-In:** For signing in, the user enters their username and password, which are compared to stored values. If they match, access to the diary system is granted; otherwise, an error message prompts the user to try again.
* **Managing Diary Entries**

The core functionality of the system is diary entry management. The user can add, view, edit, or delete entries. Each entry is encrypted before

**2. SYSTEM ARCHITECTURE AND DESIGN**

The Personal Diary Management System has been designed with a clear and modular structure that ensures easy management of user data, secure privacy for diary entries, and a seamless user experience. Below is a breakdown of the core components of the system, each of which plays a specific role in its overall functionality.

* **User Authentication System**: The system incorporates a user authentication feature to secure the access to the diary. When a user registers, they provide a unique username and a password, which is encrypted for storage. This ensures that personal information remains protected. During sign-in, the system verifies the entered username and password against the stored data to authenticate the user, granting access only if the credentials are correct.
* **Diary Entry Management**: This module handles the creation, editing, viewing, and deletion of diary entries. Entries are stored securely through encryption, ensuring that sensitive information remains private. When users add or modify an entry, the content is encrypted and saved. To view or edit an entry, the system decrypts the information for authorized use. This feature guarantees that users can safely maintain their private thoughts without external access.
* **Data Storage and File Handling**: Data is stored locally in a text file named according to the user's username (e.g., username\_diary.txt). This isolated file system ensures that each user's diary is kept separate and secure. The program supports operations such as reading existing entries upon login and writing new entries after edits or additions. The system is designed to manage file-related errors effectively to prevent data loss.
* **Security and Encryption**: A crucial aspect of the system is its encryption mechanism, which safeguards diary content. The encryption process is applied to every entry created by the user. If a user adds, modifies, or views an entry, the content is encrypted to protect the privacy of their data. This ensures that even if the file is accessed by unauthorized individuals, the contents of the diary remain unreadable. Only the registered user is able to decrypt and access the entries.

The entries are read from the file into an internal array. This allows the system to access the entries whenever needed.

* Saving Data: Every time a new entry is added, edited, or deleted, the system updates the file. This ensures that all changes made by the user are permanently stored.
* **Encryption and Decryption**

To ensure privacy, every diary entry is encrypted before being written to the file. This encryption step ensures that the diary content remains private, even if someone gains access to the file. A basic encryption method, the XOR cipher, is employed in this implementation.

* **Encryption:** Before saving an entry, it is encrypted using XOR encryption. This method alters the characters in the entry by XORing each character with a pre-determined key, rendering the text unreadable without decryption.
* **Decryption:** When displaying an entry, the system decrypts the content by reversing the encryption process. XOR encryption is reversible, so applying the same key again restores the original content.
* **User Interface**

The system operates with a simple, text-based interface, which provides the user with an easy-to-navigate menu. This interface is designed for simplicity, guiding the user through each available option such as registering, signing in, managing diary entries, and logging out.

The menu options are clear and concise, with the user prompted to select an action. For example, a user can add a new diary entry, view their existing entries, or edit and delete any particular entry.

* **Security Considerations**

Although the system uses encryption for diary entries, there are several areas that can be improved for better security. Currently, the passwords and diary entries are encrypted using basic methods. Future improvements could include:

being stored and decrypted when the user wishes to view it, maintaining privacy. The system uses a simple XOR encryption method to ensure that the content remains unreadable without decryption.

* + **Adding Entries:** Users are allowed to add entries, each of which is encrypted before being stored. The input is validated to ensure it does not exceed the maximum length.
  + **Viewing Entries:** The system provides an option to view all previous entries. The entries are decrypted when displayed, allowing the user to see their content.
  + **Editing Entries:** Existing entries can be modified by selecting the appropriate entry number. Once edited, the content is encrypted again and saved to the file.
  + **Deleting Entries**: The user can delete any entry they no longer wish to keep. After deletion, the entries are re-arranged, and the file is updated.
  + **Deleting All Entries:** If the user wishes to clear all entries, they can use this option, which resets the diary file to an empty state.
* **File Handling**

File handling is essential for data persistence. The user's diary entries are stored in a separate text file, named after the user (e.g., username\_diary.txt). When the program runs, it reads the file to load the entries into memory. After each modification, the file is updated to reflect the latest changes.

* + **Reading Data:** On user login, the program loads the user’s diary file.

* **User Authentication**

The registration and sign-in processes worked smoothly. Users were able to register with unique usernames and passwords. The system handled login attempts by validating credentials correctly. If a user entered incorrect login details, the system prompted for a retry, preventing unauthorized access.

* **Diary Entry Management**

Adding Entries: Users could add diary entries up to 100 characters. Entries were encrypted automatically before being saved to ensure privacy.

* **Viewing Entries:** Entries were retrieved from the file and decrypted successfully, ensuring users could view their entries without issues.
* **Editing Entries:** Editing functionality was fully operational. Users could modify existing entries, which were then re-encrypted and saved.
* **Deleting Entries:** Deleting individual entries worked as expected, and the diary was updated to reflect these changes. Additionally, users were able to delete all entries, resetting the diary to an empty state.
* **File Handling and Persistence**

Diary entries were stored in text files, which were loaded during login and updated after any changes. This allowed data to persist between sessions. The system effectively managed file operations, saving and loading entries accurately with no data loss.

* **Encryption and Security**

The system employed a basic XOR cipher for encrypting diary entries. This ensured that the entries were stored securely in the text file. The decryption process allowed users to view their diary entries in plain text when needed. While the XOR encryption provided basic security, it could be enhanced by implementing more secure algorithms in future versions.

* **Usability**

The text-based user interface was intuitive, with clear prompts guiding the user through each operation. Whether adding, viewing,

* **Password Hashing:** Instead of storing passwords as plain text, a more secure approach could be implemented, where passwords are hashed and salted to prevent unauthorized access.
* **Stronger Encryption:** A more advanced encryption algorithm, such as AES (Advanced Encryption Standard), could replace the XOR cipher, offering better protection against potential vulnerabilities.
* **Limitations and Future Improvements**

While the current system provides basic functionality for managing a personal diary, there are several potential improvements:

* **Graphical User Interface (GUI):** The system currently relies on a command-line interface. A GUI could make the system more accessible and user-friendly, especially for those unfamiliar with text-based interfaces.
* **Data Backup:** A backup mechanism could be introduced to ensure that users don’t lose their entries if a system crash occurs. Backups would provide additional security for the user's data.
* **Multi-User Support:** The system could be enhanced to allow multiple users to store separate diaries within the same application. This could involve a central login system to manage different user profiles.

**4. RESULTS**

The Personal Diary Management System was tested to ensure proper functionality across user authentication, diary entry management, file handling, and encryption. The following outcomes were observed:

* **Control Structures:** The project provided insight into the effective use of if-else statements, for loops, and while loops to create conditional logic for adding, editing, and deleting entries. This improved our understanding of program flow and decision-making in C.
* **Functions and Modular Programming:** By dividing the program into smaller, manageable functions such as addEntry(), editEntry(), and viewEntries(), we reinforced the importance of modular programming. This approach made the code more readable, maintainable, and scalable.
* **Encryption Basics:** We explored basic encryption techniques to ensure the privacy of diary entries. Understanding how to implement simple character-shifting algorithms helped us grasp the foundational concepts of data security and encryption.
* **Memory Management:** Even though the project used fixed-size arrays, we became more familiar with memory allocation and management in C, especially when handling strings and arrays dynamically. This provided a foundation for understanding memory-related concepts for future projects.

Overall, this project helped us develop a deeper understanding of C programming by applying fundamental concepts in practical scenarios, providing valuable hands-on experience in areas such as file handling, encryption, and data management.

**6. CONCLUSION**

The Personal Diary Management System successfully meets the objectives of providing a secure and user-friendly platform for managing personal diary entries. Through the use of basic file handling, users can add, edit, view, and delete entries with ease. The integration of encryption ensures privacy, while the system’s performance remained stable throughout testing.

Although the system is functional, several areas for improvement have been identified, such as the enhancement of encryption algorithms, the implementation of password hashing, and the possibility of adding multi-user support. Additionally, the system could benefit from a graphical user interface (GUI) to improve user interaction.

editing, or deleting entries, users were able to complete tasks without confusion.

* **Limitations and Areas for Improvement**
* **Password Security:** The system stored user passwords in plain text. For improved security, password hashing and salting could be implemented.
* **Encryption:** The XOR encryption was a basic solution. For stronger security, more advanced encryption algorithms like AES could be used.
* **User Management:** Currently, only one user can access the diary at a time. The addition of multi-user support could make the system more versatile.
* **Graphical User Interface:** A graphical user interface (GUI) could enhance user experience and make the system more accessible.
* **Performance**

The system performed well under normal usage, with no delays or crashes when handling diary entries. The data storage and retrieval process was fast enough for personal use, even with a significant number of entries.

1. **LEARNINGS FROM THE PROJECT**

Through the development of the Personal Diary Management System, several key concepts in C programming were explored and strengthened:

* **File Handling:** We learned how to read from and write to files in C, which is crucial for storing and retrieving user data persistently. The ability to manage files through the fopen, fclose, fgets, and fprintf functions allowed us to implement diary entry saving and retrieval functionality.
* **String Manipulation:** C’s string manipulation functions like strcpy, strlen, and strcmp were vital in handling usernames, passwords, and diary entries. We gained practical experience in managing fixed-size character arrays and performing various operations to ensure accurate data storage.

In its current state, the system serves as a reliable solution for users looking for a simple and secure way to maintain a personal diary. Future enhancements could significantly improve its security, usability, and overall functionality.

**7. ACKNOWLEDGMENT**

I would like to express my sincere gratitude to all those who supported and contributed to the completion of this project. I am thankful for the opportunity to work on this project and gain valuable experience in C programming, which has significantly enhanced my skills.

Special thanks to my friends and peers who provided insights and feedback during the development process. Their encouragement and suggestions helped me overcome various challenges and improve the overall quality of the system.

Lastly, I would like to acknowledge the resources and documentation available online, which provided essential information and guidance for implementing key features of the diary management system.