ENIGMA

Messaging through encryption and decryption

Group members:

Muhammad Hamza	Abdul-Mannan Ibrahim	Hafiz M. Zarar
23-NTU-CS-1186	23-NTU-CS-1003	23-NTU-CS-1030

Section	BSCS-3rd-A
Course Name	Data structures and Algorithm
Submit to	Dr. Salman, Mr. Abdul Basit
Date	31-12-2024

Contents

1. Problem Statement 2
2. Solution 2
   - Admin Management: 2
   - User Messaging System: 2
   - Encryption & Decryption: 3
     - Caesar Cipher: 3
     - Reverse Cipher: 3
   - Secure Authentication: 3
3. Data Structures Used 3

   Linked List: 3

   Stack: 3
   - Encryption Process: 4
   - Decryption Process: 4
4. Functionalities 4

   Admin Functions: 4
   - Login: 4
   - Create New Users: 4
   - Edit User Passwords: 4
   - Delete Users: 4
   - View User List: 4

   User Functions: 5
   - Login: 5
   - Send Messages: 5
   - Receive Messages: 5
   - View Sent/Received Messages: 5

   Encryption and Decryption: 5
   - Caesar Cipher: 5
   - Reverse Cipher: 5
5. Sequence Diagram & Flowchart 6
6. Storage: 6
6. Conclusion 7

1. Problem Statement

In today's digital world, communication within an organization plays a vital role in maintaining smooth operations, improving productivity, and ensuring coordination among team members. However, many organizations face the challenge of securing sensitive internal communication from unauthorized access. Simple messaging systems may not offer the necessary protection against potential data breaches or unauthorized access.

This project aims to address these challenges by creating a secure messaging system specifically for organizations, where communication is restricted to within the organization. The system will allow users to send messages to each other while ensuring that these messages are protected through encryption. Additionally, it will allow administrators to manage user accounts with essential functionalities such as creating new users, editing passwords, and deleting accounts.

The primary objective of the system is to provide secure communication where only the sender and the intended recipient can decrypt and read the messages. To achieve this, a combination of two encryption techniques, **Caesar Cipher** and **Reverse Cipher**, will be used. These ciphers, while basic in nature, will ensure that messages cannot be easily understood by unauthorized individuals, ensuring that privacy is maintained. The problem, therefore, involves implementing a secure and efficient messaging system with user management capabilities, providing both security and ease of use for all members of the organization.

2. Solution

The solution to the problem is a C++-based application that provides secure messaging between users within the same organization, with additional administrative control for managing user accounts. The main components of the solution include:

- **Admin Management:** Admins have full control over user accounts within the organization. This means that they can view, create, edit, and delete user accounts. However, admins do not have the ability to send or receive messages within the system. Their role is purely administrative.
- **User Messaging System:** Users can log in to the system and send encrypted messages to other users within the same organization. The messages sent by users will be encrypted using a combination of Caesar

Cipher and Reverse Cipher to ensure that only the intended recipient can decrypt and understand the message.

- **Encryption & Decryption:** The encryption system involves the use of two classical ciphers:
    - **Caesar Cipher:** This cipher shifts each letter of the message by a certain number of positions in the alphabet. For example, if the shift value (key) is 3, the letter "A" would become "D," "B" would become "E," and so on. This provides basic encryption.
    - **Reverse Cipher:** After applying the Caesar Cipher, the entire message is reversed. This adds an additional layer of encryption and further obfuscates the original message. Only the intended recipient, who knows the decryption key, can reverse the process and retrieve the original message.
- **Secure Authentication:** Only authenticated users of a specific chosen organization can log into the system and access the messaging features. Authentication will be done using usernames and passwords, ensuring that only authorized individuals can access their accounts.

The system will utilize the **Linked List** data structure to store users and admins information, allowing efficient management of dynamic data.

3. Data Structures Used

Linked List:

In this project, the **Linked List** data structure is used to manage users and messages dynamically. This data structure is ideal for handling operations like viewing, creating, editing, and deleting user accounts. Here's how the linked list will be applied:

User Linked List: Each user's information (username, password, organization) is stored in a node of the linked list. The admin can traverse this list to view users' info, create new users, modify existing users, or delete users from the system. The linked list allows for efficient dynamic memory allocation and easy insertion or deletion of users.

Stack:

The **Stack** data structure is specifically used for the **Reverse Cipher** part of the encryption process. Since a stack operates in **Last In, First Out (LIFO)** order, it is ideal for reversing the order of characters in a message.

The Reverse Cipher Implementation through stack happens two times during the execution of the program:

- **Encryption Process:** Each character of the message is pushed onto the stack. The characters are then popped from the stack and appended to a new string, effectively reversing the order of characters.
- **Decryption Process:** The encrypted message is processed in reverse, where each character is pushed back onto the stack. Characters are popped off in reverse order, reconstructing the original message.

The stack is well-suited for this task because it processes elements in reverse order, which matches the goal of the Reverse Cipher. This structure provides a clear and efficient method of achieving the cipher's encryption and decryption operations

4. Functionalities

The application will offer several key functionalities for both admins and users:

Admin Functions:

Admins will have full control over the management of users within the system. Their specific tasks include:

- **Login:** Admins must authenticate themselves by entering their organization's code and password. Only valid admins with correct credentials will be allowed to access the admin system within the organization.
- **Create New Users:** Admins can create new users by entering a username, password, and other necessary details. When a new user is created, their information will be added to the user linked list.
- **Edit User Passwords:** Admins can modify the password of any user in the system. This feature is useful in case a user forgets their password or requests a password change for security reasons.
- **Delete Users:** Admins can remove users from the system. Deleting a user will remove their information from the user linked list, ensuring that they can no longer log in or access the messaging system.
- **View User List:** Admins can view a list of all users in the system. This helps keep track of user accounts and their status within the organization.

It is important to note that while admins have control over user accounts, they do not have the ability to send or receive messages. Their role is focused entirely on user management.

User Functions:

- **Login:** Users must authenticate themselves by entering their username and password. Only valid users with correct credentials will be allowed to access the messaging system.
- **Send Messages:** After logging in, users can send messages to other users within the organization. The messages will be encrypted using the combined Caesar Cipher and Reverse Cipher method, ensuring that unauthorized users cannot easily read the contents.
- **Receive Messages:** Users can receive encrypted messages from other users. Once the message is received, the user will apply the decryption process (reverse cipher followed by the Caesar cipher) to retrieve the original message.
- **View Sent/Received Messages:** Users can view their sent and received messages, which will be stored in the message linked list. This provides a simple way to manage past communications.

Encryption and Decryption:

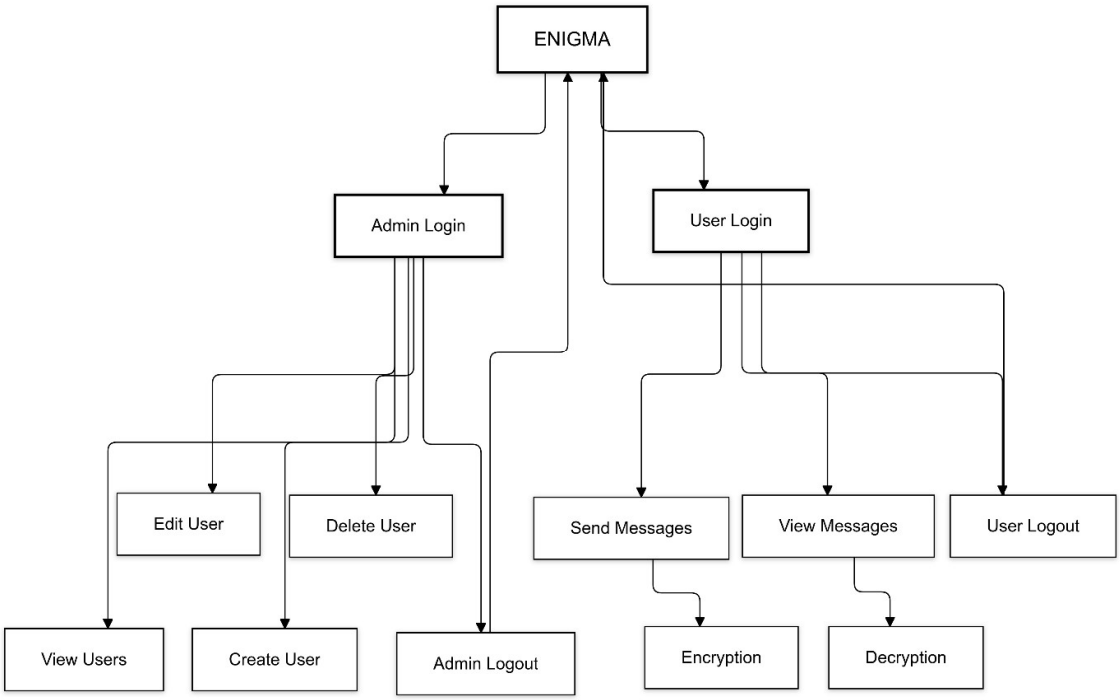
The encryption process will be applied to every message sent between users. The combination of **Caesar Cipher** and **Reverse Cipher** ensures the following:

- **Caesar Cipher:** Each letter in the message is shifted by a specified key value (e.g., 3 positions). This transforms the original message into an obfuscated version.
- **Reverse Cipher:** After applying the Caesar Cipher, the entire message string is reversed, further obfuscating the message and increasing security.

When the recipient receives the encrypted message, they apply the reverse cipher first and then the Caesar cipher to decrypt it, ensuring that only they can understand the original content of the message.

5. Sequence Diagram & Flowchart

To understand the interaction between the components of the system, the following diagrams can be used:



6. Storage:

For storage of the data, file handling is used. The usernames and passwords are stored in a file named by that organization and the messages along with the sender and receiver usernames are saved in another file names as organization name + "_messages.txt".

For storing usernames of the organizations (here army, education, bank, and business), these files are used:

- army.txt
- education.txt
- business.txt
- bank.txt

For storing the messages within these organizations, these files are used:

- army_messages.txt
- education_messages.txt
- business_messages.txt
- bank_messages.txt

Our system can manage dynamic creation of the messages file, we just have to modify the 'main function' and the rest of the work is done within the 'Enigma' class dynamically. It means, one can add organizations easily and hence our system provides flexibility for addition of organizations.

6. Conclusion

This project aims to provide a secure and efficient messaging platform for users within an organization. By using classical encryption techniques like Caesar Cipher and Reverse Cipher through **Stack**, the system ensures that sensitive communications remain private. The admin and user functionalities are clearly separated, allowing for effective user management and secure communication. The use of the **Linked List** data structure for storing users enables dynamic memory management, making the system scalable and efficient. The overall goal is to create a system that balances simplicity, security, and functionality, providing an effective communication tool for organizations. Furthermore, we can add more organizations easily with barely minimum changes in our code.