SafeSecretShare

Have you ever given your password or API credentials to someone else to allow them temporary access to your accounts? If so, you're not alone. However, it's important to consider the security implications of sharing sensitive information. Ideally, passwords should only be shared with the intended person for a limited time, such as a day or a week. The main security concern is that if our communication channels (such as email, iMessage, or WhatsApp) are compromised, the attacker could gain access to other credentials.

To address this problem, I am planning to create a platform called SafeSecretShare that will allow users to securely share sensitive information such as passwords, API credentials, and authorization tokens. The platform will not store any sensitive data in plain text, but instead will use strong encryption to protect the information. The encryption key will only be shared with the owner, who can then decide who they want to share it with. This will enhance the overall privacy of the transaction.

The main features of SafeSecretShare will include

- Ability to share sensitive information securely
- Store data in encrypted format
- Allow users to set time limits for automatic deletion of data
- Limit the number of times data can be viewed
- Provide users with an access log for their sensitive data

References:

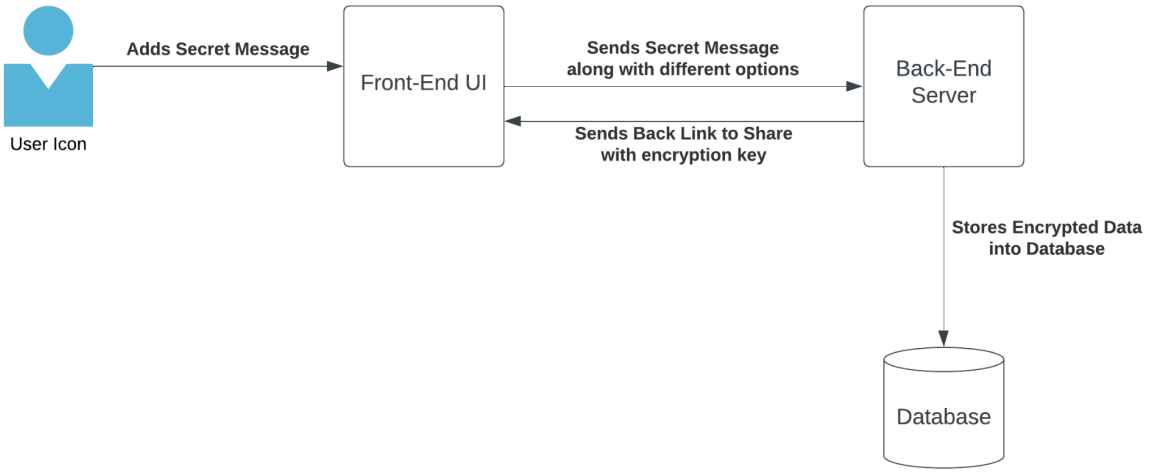
1. https://wormhole.app/

Technical Details:

Front-end technologies: React, HTML, CSS, JavaScript
Back-end technologies: Golang, Docker

Database: MySQL or Postgres
Cloud Technologies: AWS KMS, AWS Aurora DB, AWS EKS

Block Diagram:



API Details:

- **/api/v1/post-message**: Takes secret messages from the user and encrypts the data, stores it in the database and returns the encryption key with a message unique UUID.
- **/api/v1/get-message:** Takes encryption key with message unique UUID and returns the plain text data.
- **/api/v1/get-access-logs:** Takes messages unique UUID and returns the access logs associated with the message.

Future Features:

- Allow users to share any message such as image, video, file etc.
- Allow users to choose different encryption algorithms based on their security awareness

UPDATE 05/26/2023:

We need to make changes to current architecture. The current architecture relies on a back-end server to generate encryption keys. This would lead to security issues in case the server gets compromised. The more secure way to design the system is to create encryption keys on the client side and only send encrypted messages to the backend. This would ensure confidentiality of messages even in case of server compromise as the server does not have encryption keys to get back confidential messages.