End-to-End Encrypted Chat

abconversation.us

Visal Hok and Jacob Parcell

Requirement

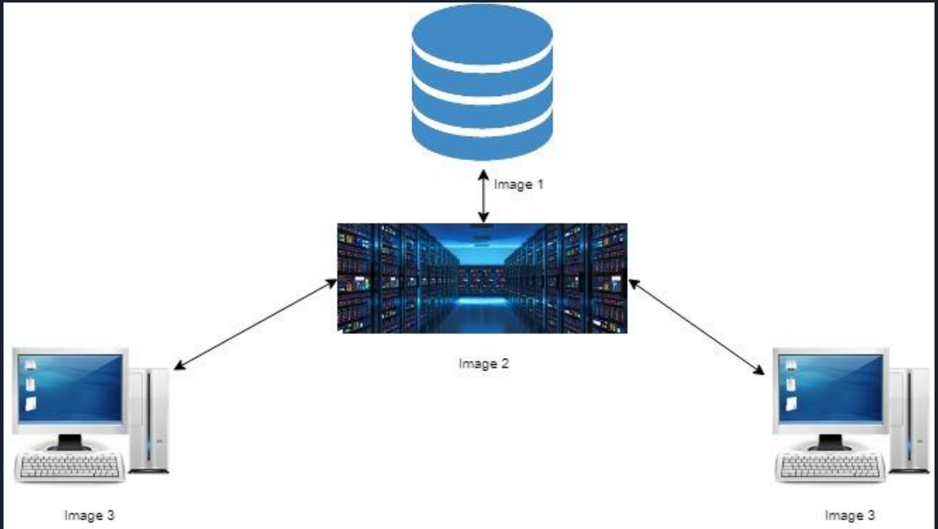
The purpose of this application is to provide end-to-end encryption for one-to-one messaging. In order for our application to be end-to-end, the application must be designed in a way that protects the user from the server as if the server was an adversary. This means we cannot trust the server with the distribution of keys and we also cannot allow the server to be able to understand messages going between users.

Design

- ASSETS
  - Username
  - Password
  - Messages
  - RSA Private Key
- STAKEHOLDERS
  - User
  - Application Owner
- ADVERSARIES
  - Active/Passive Insider
  - Active/Passive Outsider

Design

- Client
- Server
- Database



Solution

- User sent their RSA Public Key through email
- Messages are encrypt and decrypt on the client using AES and HMAC for message integrity
- AES Key and HMAC key are encrypted using the receiver RSA Public Key
- The server is uses for users authentication and authorization as well as messages transfer
- The client side requires the sender's public key before displaying their message to protect against unauthorized messages

Implementation

- Encapsulation/Decapsulation
- Public Key Exchange through Email
- Sign-up
- Login
- Send Message
- Receive Messages

Future Work

- User Remote Login
- Forward Secrecy
- Better GUI
- Automate email key exchange
- Group chat

Image Citation

Image 1: http://clipart-library.com/clipart/database-free-download-png.htm

Image 2: https://blogs.technet.microsoft.com/uktechnet/2017/01/17/how-to-boost-your-windows-server-2016-security/

Image 3: https://upload.wikimedia.org/wikipedia/commons/1/1a/Crystal_Project_computer.png