End-to-End Secure Chat

Team Insecurity: James Dinh and Hsi You Huang

Application Properties

1. **Features**: The chat is a one-on-one system, with a middleman. The two "participants" are the clients. The middleman is the server. One client sends a message to the server; the server receives that message and passes it on the other client; the other client receives the message. The clients do not have to be active at the same time in order to receive messages.
2. **Language**: Java
3. **Platform**: Web app, Amazon AWS, Spring framework for Java, AES-256 encryption.
4. **RESTful server**:
   - The server will manage each User of the chat. Each user will have its own ID and ?
   - POST will create new User with a new ID and ?.
   - GET will read the information of user based on ID.
   - PUT will update info on each user.
   - DELETE will delete the user from the server.
   - Above functions work with HTTP.
5. **Assets**
   - Client
   - Server
   - User information
   - Message
6. **Stakeholder**
   - Users of the chat system
7. **Adversarial Models**:
   - Outsider: Outsider can be passive or active. Active attacker tries to temper network traffic to intercept communication. Passive attacker eavesdrop on the communication.
   - Insider: Insider have access to our communication network, and can pretend to be one of the client to the server, or temper with the communication channel to their advantages.

8. **Possible Vulnerabilities**:
    - Man-in-the-middle attack
    - Brute force attack
    - Work station hijack
    - Encryption cryptanalysis
    - Keylogger
    -

9. **Related Previous Work**:
    - WhatsApp
    - Signal

10. **Solution:** The clients will hold a pair of keys that will encrypt and decrypt messages on their end. Example: Sender client sends a message to its receiver client. Sender will type up their message and "send" it. Before the message leaves the sender, the message is encrypted and then it gets sent to the server. The server will receive the encrypted message and then passes the message along to the receiver. The receiver will receive the encrypted message, decrypt it, and then the client "receives" the message (they are able to read it).
    Against the man-in-the-middle attack, TLS has certificates available for use. We can use the certificates to authenticate the two clients. Using this authentication, we can prevent man-in-the-middle attacks.
    Against brute force attack, we will use AES 256 bits encryption to encrypt the messages. To brute force an AES-256, it would take far too much money and energy, and far too long to be worth while. For this reason, AES-256 would achieve computational security for this project. Using AES-256 also provides solution to the cryptanalysis vulnerability, as there are no known cryptanalsys against it.

11. **Analysis**:
    - Confidentiality
      The AES encryption will offer strong protection against an outside adversary attempting to gain access to the client's messages.
    - Integrity
      Against an insider man-in-the-middle adversary, certificates will be used to verify the identify of the message source. If a man-in-the-middle (MITM) adversary is detected, they will be unable to receive the message and any messages sent by the MITM will not be received by the clients.

- Authentication
  The certificates will provide a form of authentication. Messages received by a client will have its certificate checked. If it passes, the message is confirmed to have been sent by the other client. If the certificate doesn't pass, the message is not confirmed to be from the other client.