End-to-End Secure Chat

Team Insecurity: James Dinh and Hsi You Huang

**Application Properties**

* **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.
* **Language**: Java.
* **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.

**Assets**

* Client
* Server
* User information

**Stakeholders**

* Users of the chat system

**Adversarial Model**

* Eavesdropper

**Possible Vulnerabilities**

**Related Previous Work**

**Solution**

The clients will hold a “key” 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).

**Analysis**