Nolan Burfield
Assignment 4
CPE 401
5 May 2015

Running

First setup the database for the server to run on MySQL

Run these commands in MySQL:

```
# mysql -u root -p
# CREATE USER 'social_peer'@'localhost' IDENTIFIED BY 'password';
# CREATE DATABASE social_peer_db;
# GRANT ALL ON social_peer_db.* TO 'social_peer'@'localhost';
# exit
```

Then from the server directory run:

```
   python database.py
```

* This will need peewee installed

** In database.py there is the data used to access the database **

Running the App on emulator.

To Run the app running on a emulator will require port forwarding from the selected Peer Port to the emulator

To exit from the server side hit control-c.

The app will run all the server commands, and will run a HI message, and Chat messages.

The application should be intuitive with the commands.

To do a Chat there will be a list of friends, and then type in the friend name initially to chat to. This will then change for a message to be sent. Type that in, and then the message will be sent b UDP to the receiving friend.

Security

To add in the security to the android application there was a few things that were done. To start off there is the need for the user to know the public key of the server, and this information is stored in the android application. When sending messages from client to the server the packets are encrypted with an RSA algorithm. What must be done is the server public key in string format is encoded, then added to a

Java PublicKey variable. With the Java variable the packet is encrypted on the client side, and sent to the server. On the server side the packet is received, and must now be decrypted. The server reads the private key from file and decrypts the message; this establishes confidentiality with the server and client.

In order to have authentication of the client the packets sent between the two entities must be encrypted with the sender private key. On the server side the user will have a public key stored in the database in order to authenticate the client packets. Packet encryption is done by first encrypt with client private key followed by the server public key. With responses from the server it will require the opposite; encrypt first with the servers private key followed by the client public key. This method will create confidentiality and authentication.

Communication between two peers will require security as well. Initial communication in secured by the receivers public key, which is acquired from the search query to the database. When a communication between two peers is greater than one packet a symmetric key is established. This key is to be encrypted by the connection-establishing peer and sent to the other peer. The packet with the symmetric key is encrypted with both the sender's private key and the receiver's public key. This method will again enable the two parties to know confidentiality and authentication.

A replay attack is also considered with replay sensitive packets. Chat is a good example of replay sensitive since it involves data being viewed by the users. In order to handle this a counter is established that will not allow the same chat number to be read. This number is secure due to the encryption of the packet by the established symmetric key.