**Project P5 – Design Document**

Nick Forquer, William Greer, Gerard Puhalla, Trevor Rambacher

**General Description:**

As stated in the project update, the group has implemented an app that gives the user the option to send a file via WiFi-direct or Bluetooth connection. We have a file set up that will be our test document for transferring on both the WiFi-direct and Bluetooth connections. The app sets up a connection between two devises utilizing the selected communication format. Once this connection has been established, both devises begin performing the Diffie-Hellman algorithm by creating public keys that they send to one another and private keys that they use to generate the secret shared key. After both sides have calculated the secret shared key, they can send encrypted messages across the connection. When they receive a message, they know it came from the other party and can easily decrypt it to read the message. Aside from establishing the connection between phones and sending messages between the phones, the two connection types use the same Diffie-Hellman class and encrypt/decrypt the messages.

**Security Approach:**

Before we even start to connect devices, we need to obtain eight different permissions, three for Bluetooth file transfer and three for Wifi-direct, and two for both. For Bluetooth, we need both Bluetooth and Bluetooth Admin permissions as well as accessing location in order for the devices to locate each other and set up a socket connection. For Wifi-direct, we need the ability to access and change the wifi state in order to setup a connection, as well as being able to access the network state. The last two permissions that we need to obtain involve the reading and writing of external files, since users will want to send their personal files taken from outside of the application.

In terms of our security approach, we utilize Diffie-Hellman key exchange to encrypt the file using AES encryption. This is utilized at the application layer as this seems more practical than hacking the phone’s Bluetooth stack. Before sending the files through the sockets set up for Bluetooth, we encrypt the file before we send the file. The key parts are send in the clear beforehand, and although this is a potential weakness, a man in the middle attack isn’t feasible at this point with the three step connection used by Bluetooth to connect the receiver to a sender before we even get to the key exchange step. This three step connection involves first the receiver setting up a server socket, a sender connecting to the receiver using a MAC address that they receive from the receiver, and lastly both transitioning to a socket connection that no other parties can enter. A man-in-the-middle attack, however, can occur fairly easily at this step with the adversary by either obtaining the MAC address of the receiver and beating the sender to the connection or convincing the sender to use their MAC address.

**Screenshots of Design and Functionality:**

Below are some screenshots of what our application looks like and several screenshots that show the file transfer process for both Bluetooth and Wifi-direct.