M2 - Sprint Report

Activity breakdown

Max:
Initially, I worked on creating general cryptographic functions that would be used for all of our cryptography(~2 hours). Next, I added a handshake functionality that would share a session key between two users and store them, and also verify the integrity of the message (~6 hours). I then added an AES encryption to each message using the two keys hashed from the original key, one for encrypting and one for creating tags with an HMAC function (~4 hours).

Pavle:
Initially, I worked on Gen script and on initial database code with Stefanos who then later took over completely. (~2 hours) Next, I worked on our initial version of Requests module that we later adopted a little bit as we gained a better idea of what our client and encryption module did. (~2 hours) We as a team then worked on sending and receiving handshakes as well as on refining our client and server scripts. (~8 hours)

Stefanos:
Initially, I worked on Database script with Pavle and then later on I added logging functionalities and tested all the functions in the database (~3 hours). Then we as a team worked on our handshake protocol which was the task that took us the longest time (~6 hours). Together we also made some changes to the client and the server code outside sending and receiving handshakes that had to do with adopting our existing code to Request module we added after we realized that sending and parsing data as comma-separated strings will not work for our use case. (~2.5 hours)

Addy:
Initially, I worked on the Client script and Server script to set up a basic chat (without encryption) between clients. (~3 hours)
Then, as a team we spent roughly 8 hours working on sending/receiving handshakes, refining client/server scripts, and encrypting messages for clients.

Productivity analysis

We completed a bit more than we intended. For our initial plan for our alpha version, we wanted to create a simple chat interaction between two clients through a server and that the message would be encrypted in some way. We ended up creating a handshake between two clients that want to chat, using AES encryption with RSA key pairs, as well as using MAC to sign the message, preserving integrity as well. We also got a headstart in our database which we will use as server logs.