Weekly Blog 12

This was the final week of our App development. We were able to successfully code, integrate and test our app for validity. Some of the main things we did this week are as follows:

As explained in my previous weekly blog 11, we had successfully tested our various scenarios. Some of the scenarios are:

1. Testing at a distance of 1.5 metres or social distancing criteria
    a. Method:
        i. 2 devices for 2 people
        ii. Test subjects 1 and 2 held their phones in hand
        iii. An infected person could be anyone amongst the two
        iv. The app should not notify infected due to social distancing.
    b. Result:
        i. The app was successful in avoiding a false risk notification while maintaining social distancing.
2. Seated scenario:
    a. Method
        i. 2 devices for 2 people
        ii. Test subjects 1 and 2 held their phones in hand
        iii. Each run was for 5 minutes
        iv. Test subjects did not change their positions
    b. Result:
        i. App successfully notified the contacted person with exposure notification
3. Phones in the pocket scenario:
    a. Method
        i. Test subjects 1 and 2 had phones in the pocket.
        ii. The test was run for 2 mins.
        iii. Subject 2 was infected and uploaded his keys
        iv. Subject 2 downloaded keys and checked for notification
        v. Subjects did not change positions
    b. Result
        i. App successfully notified the contacted person with exposure notification
4. Test for false positives:
    a. Method:
        i. Infected and non-infected users are separated by a concrete wall.
        ii. Test subjects do not change their positions for 5 minutes.
        iii. Phones could either be on hand or in the pocket.
        iv. False-positive if the app detects infection even with the separation of a concrete wall.

b. Result:
    i. App notified a false risk notification to the person on the other side of the wall.
    ii. Possible verdict: reduce the TxPower of signal, but however, the app still notified the users.
    iii. Causes false positives.

Apart from this, we had run MobSF for analysing the security of our app. We received a score of 6.5 which meant medium risk. We had resolved some of them based on MobSF report.

Some of the implementational changes I had done was removing all toast messages and replacing them with Alert boxes for better usability.

Commits:

1. https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/commit/0a2d2822977cce7626a329c73848398e2df2f2d8
2. https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/commit/0eff9456d79732300633717dc3e81a3614e6928f

Apart from this, we have created the video with various test scenarios and app/server functionalities explained. YouTube link: https://youtu.be/N1ccFrFehd8

Conclusion:

Overall, we were successful in implementing a contact tracing app with a focus on privacy-preserving architecture. Our app meets the minimum performance, privacy and security specifications which we have analysed and included in our presentation. We were successful in developing the Google/Apple Exposure Notification API from scratch and incorporate it as part of our App.

Acknowledgments:

We would like to thank our professor, Damith Ranasinghe and our supervisor, Zach Wang for their consistent support and guidance in making this project happen.