CS 573 – Final Exam Solution

Report 1 :

https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ContactTracing-BluetoothSpecificationv1.1.pdf

Report 2 :

https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ContactTracing-CryptographySpecification.pdf

Report 3 :

https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ContactTracing-FrameworkDocumentation.pdf

LinkedIn Article :

https://www.linkedin.com/pulse/tutorial-contact-tracing-infrastructure-edward-amoroso/'

After reviewing all of the three reports on the Apple/Google Contact Tracing protocol, including the LinkedIn post on the given topic (references of the links given above), I have summarized some details explained in the following paragraphs.

Solution:

As explained in report 1, contact tracing is being supported by contact detection which is based on BLE (Bluetooth Low Energy) service registered with the Bluetooth SIG with 16-bit UUID 0xFD6F and is designed to enable proximity sensing of *Rolling Proximity Identifier (RPI)* between devices for the purpose of computing an exposure event. Now, talking about Bluetooth, the first thing which comes into mind is the frequency and the range of the Bluetooth device. Contact tracing is used to decide how long the entitled bodies were in contact and how far they were from each other**.** The contact detection service will sense the RPI. Now, report 2 explains that RPI will contain a timestamp which will help us to track the time of contact but measuring the distance between the devices (people) still remains a question. One more disadvantage of using Bluetooth for contact tracing is that Bluetooth uses electromagnetic waves that can pass through walls. Now, let's suppose if there are some people and a person (who was later diagnosed with Covid-19) are separated by a wall then it's hardly useful to warn the rest of the people that they were exposed to a Covid-19 diagnosed person. In my opinion, a possible solution to the above scenarios is to track the signal strength while detecting the identifiers and storing that information in a property in **CTContactInfo** (report 3). Now, if there is a wall between devices, the signal strength will drop, and some function could be implemented in **CTExposureDetectionSession** (report 3) to use this signal strength and determine the distance between the users and sending data through **CTExposureDetectionontactInfoCompletion**.

In report 1 and report 2, the privacy portions explain that Contact tracing and the server operator implementing this protocol does not hold the location information of the users. In that scenario, as explained in "*Matching Values from Users Tested Positive*", if the list of *Diagnosis Keys* is being fetched frequently then there would be a lot of data considering the huge amount of people diagnosed in a week or so, that would be stored locally on the phone and that can create storage issues. A possible solution to this scenario is to send an encrypted IP address with the *Diagnosis Key* on the server when a person is diagnosed with Covid-19 and then the server decrypts it to decide which IP addresses should receive this alert. To execute that, we can use the *Public-key cryptography*, where the contact tracing service will have the public key to encrypt the IP address that can be sent on **CTExposureDetectionontactInfoCompletion** along with **CTContactInfo** and the private key to decrypt the IP address will belong to the server. Now, we can make arrangements in the server to isolate this decrypted IP address in a way that when client devices fetch the data, the server can process the IP address (using TCP/IP protocol) from where the request is being made and provides only the *Diagnosis Keys* based on IP address filtration. Only the point to note here is that the server should be "secure enough" to store the IP address of the diagnosed users because other than that as we are not sending IP addresses to the clients or storing any other IP address, the location of the users are secure.

In report 3, **CTStateGetRequest** is used to check if contact tracing is on or off on the device and **CTStateSetRequest** changes the state of contact tracing on the device. Now, after referring to the LinkedIn article where there is a point of "*Absolute minimal functionality*", my opinion about setting the state request is that this service should make it mandatory for every device to use contact tracing. The reason behind doing that is, first of all it will reduce the functionality and also, the service is not using any data of the user until the user decides to upload a positive diagnosis test willingly. However, I am not sure that it would help to make this service more secure but definitely in this crisis, the government and the public would like to maintain as much transparency as they can and letting the user decide to use this service or not may ruin the whole concept of using contact tracing if most of the people decided not to turn on this service.

In report 3, the contact tracing flow explains how the *Diagnosis Keys* would be sent but in none of the report's user identification is discussed. What if the user uploads *Diagnosis Keys* on just an assumption thinking that he/she has symptoms of Covid-19 and not actually performs a test or can even have symptoms of any other flu? A possible solution to the given scenario can be that the user should be provided with some unique ID (something like OTP) that can be received from an authorized health care organization after he/she is diagnosed with a positive test to Covid-19. Now, when the user uploads the *Diagnosis Keys*, the server should ask for authentication and by using the unique ID provided by the user, the server should be able to authenticate the user. This all could be possibly done while performing **CTExposureDetectionSession**. Also, to note here is that the server should be able to authenticate it securely, and possibly we can use authentication protocol like Kerberos. Now, the reason behind using Kerberos is that the user cannot rely totally on the health care organization because there are still chances of getting the data/unique ID leaked. Thus, using Kerberos authentication the identification will be secured from every end and can also avoid the situation of Brute-force attack.

Weaknesses:

After going through the LinkedIn article and relating them with the given reports, there are few points that could be suggested. The foremost point to that is phishing messages and other attacks. None of the papers discuss about the protocols used for requesting the *Diagnosis Keys* from the server. Possibly, there can be a *man-in-the-middle* attack where the attacker can provide false positive *Diagnosis Keys* and *RPI* and as described in article phishing messages/alerts could be dangerous. One more point that is highly relatable with "*Coordinated social media*" is that what if somebody leaks someone's information (maybe even with their photo) on social media that he/she received the alert message from contact tracing about a Covid-19 positive person and that alert indicates to this particular person. Now, the person may or may not be diagnosed with Covid-19 but the information is leaked on social media (this scenario is similar to somebody imitating to be someone and posting that on social media or posting fake news/posts), and that can compromise the privacy of any person.

Furthermore, the reports don't specify if the users who are using the service can access the *RPI* stored on their local devices. If that is possible, then somebody could keep track of the people they meet with time and the RPI they obtain from their device and on receiving the alert of someone being diagnosed positive they can try to relate the *Diagnosis Key* and *RPI* or the *time information* received from the server and can leak the information of that person without their concern. Moreover, after collecting the information of Covid-19 diagnosed people on a large scale, the attacker can use it to earn money by selling/revealing that information to the required organizations.

Comments and Recommendations:

- Contact tracing is really a great initiative that could help people to determine about them being revealed to some Covid-19 diagnosed people, but the privacy concerns among people are still inevitable especially with the infrastructure (as mentioned in the LinkedIn article).
- This service would help to track the transmission through contact of humans however, there are also chances of "environmental transmissions".
- The reports do not explain about the data retention policy like for how long the data would be stored on the server as well as the device.
- As explained in the "Simplified Expert Management", either there should be avoidance of language or cultural issues or this service should support multiple languages. One of the best examples for that is the "Aarogya Setu" app (for contact tracing), which is currently available in 11 different languages.
- This service should also have the latest updates and news about Covid-19 so that there can be fewer chances of false information and could also include a small guide that can help explaining initial steps to be taken, after being revealed to Covid-19 positive diagnosed person.
- In report 3, there is no explanations for **CTExposureDetectionontactInfoCompletion** (*possibly* CTExposureDetection***C***ontactInfoCompletion) that is described in Contact Tracing flow image**.** There should be a portion explaining that functionality as it contains the transmission of timestamp and duration which should be highly secured.