UniTrace: An Exposure Tracing System for Universities

Ben Steffen, Miriam Rittenberg, and Nina Gerszberg

Many deadly diseases are primarily spread through exposure to an infected person. Since people may find out that they are infected after already being contagious for some period of time, and may continue to contact other people even after testing positive, a system that tracks which people have been exposed to which other people, and who tests positive when, can help control the spread of disease. The goal for this project is to describe UniTrace, a contact tracing application that supports a university in identifying infection exposures and assists those who have been infected or exposed.

Designing a well-functioning exposure tracing system is difficult because it requires high usage rates by people in an area, requires people to frequently check whatever method of communication the system uses, and requires some method of detecting whether users are nearby. However, our system can avoid many of these issues, since it will be used in a university setup where everyone is assumed to have a phone and to have the app installed at all times.

Goals

The primary goal of UniTrace is to reduce infection. To accomplish this, the system must be useful and people must want to use it. Goals that can help achieve this are ease of use, preservation of user privacy, and providing timely and accurate responses. While still important, we put less priority on performance and providing information to researchers.

Fast notification of exposure events is one of the most important priorities, since every hour a person spends potentially infected but unaware of their exposure is an hour they may be spending infecting even more people. Similarly, ease of use is important, so that the users always see the notifications, rather than tuning out frequent emails or rarely checking the app. Since users are required to use the system, privacy is also an important consideration.

UniTrace achieves fast notification by requiring mandatory reporting of positive tests, sending out information on the positive test to every user soon after the Central Server is notified of a positive test, and having a small enough user base that checking the key pairs against contacts takes very little time.

We plan to achieve ease of use by focusing on usability when designing the app and testing the understandability of the UI before and during deployment.

The system preserves as much privacy as possible for individuals who do not test positive, and some privacy for those who do test positive. Information about who you have contacted never leaves your phone, and if you do not test positive neither the central server nor any of the other users are capable of determining your contacts. If you do test positive, anyone who comes within a short distance of you while you are supposed to be quarantining will be notified that they are near a person who has tested positive and should be quarantining. However, even if you test positive, people you have not contacted and do not contact during your quarantine period will not learn about your contacts.

Because UniTrace will be used by a university with only 20,000 people, rather than a whole country, performance is not a major issue. However, we still try to use only as much storage and bandwidth as is necessary. Also, while we hope to help researchers, that is a lower priority than preventing spread.

System Assumptions

We assume that users are tested every other day and that the testing facilities have a method of communicating with the central server used for exposure tracing. We also assume that every person is running our app at all times and will notice within a few hours if the app notifies them of an exposure event. We assume that the various modules of the system have the specs detailed below, and that BLE signals do not transmit through walls. We assume that the system is used only by around 20,000 people. We assume that we can require people to quarantine and that they will not turn off their phones to prevent the app from transmitting information.

System Overview

UniTrace consists of several components, each owned by a different party. The Smartphones are personal devices owned by the students of the university. Each phone has a personal Generator, a cryptographically secure pseudorandom number generator (CSPRNG) that it uses to generate IDs. The Smartphones exchange those IDs with other nearby Smartphones via Bluetooth Low Energy (BLE). The Routers and Central Server are devices managed by the university. The Routers forward traffic to and from the Smartphones. The Central Server stores current and historical Generators for each student, along with their status (safe, isolation, quarantine) and contact statistics. The Central Server also receives test results from medical facilities and uses those tests to determine which students need to quarantine. Using the test results, the Central Server collects the Generators belonging to infected community members, and sends them to the Smartphones for contact analysis. Finally, the Central Server also sends each Smartphone a new Generator daily. The above information is summarized in Figure 1 below.

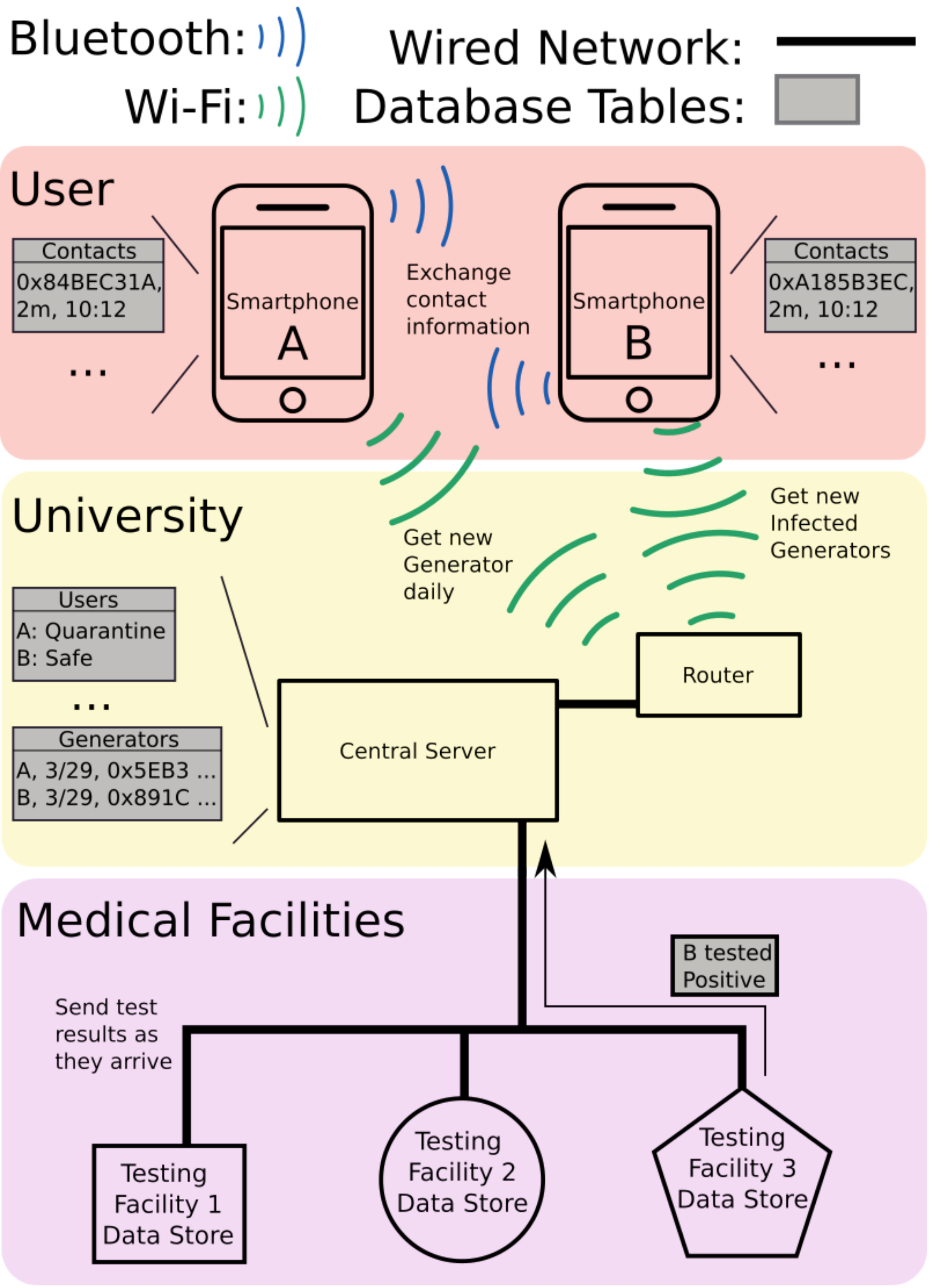

Figure 1: System overview

In this scenario, we have two users, A and B, who are in close contact with each other. They are in contact with the Central Server via the Router, which sends them the generators belonging to infected people that it has identified since their last contact, and also sends each Smartphone their new Generator daily. User B has taken a test at Testing Facility 3, which then informs the Central Server that B has tested positive.

System Description

The Smartphones

The Smartphones are handheld personal computing devices that the members of the university carry around with them as they go about their daily lives. For data storage, we use a relational database. Relational databases are very widely used and well understood, and have features providing strong reliability and durability of data storage. This reliability supports the system's ease of use, as it helps prevent the app from getting into an inconsistent state in situations like the phone running out of battery or being dropped, causing it to suddenly shut down. Table 1 below lists the database tables stored on the smartphone.

Table Name	Contents
Contacts	Timestamp, Signal Strength, Sender ID
Generators	Date of Use, Generator State

Table 1: Smartphone Database

Every 15 minutes, the Smartphones use their Generator to create a new ID which they use in the exchange of contact information. Because the Generator is a CSPRNG, it is computationally infeasible to match different IDs to the same person. This supports our goal of privacy, as it makes it very difficult for a user to deanonymize an ID and link it to a person.

Every 10 seconds, the Smartphone emits a BLE beacon that transmits the current ID to anyone in close proximity. Nearby Smartphones note this ID, along with the signal strength of the message and the time it was received in its database. The interval of 10 seconds was chosen because it results in low storage requirements on the smartphone while also not dividing the 20 minute period that constitutes an exposure event into so few pieces that it becomes difficult to accurately detect.

When a message with an ID that was derived from a Generator belonging to an infected user is received, the phone will alert the user that they have been in contact with a person who tested positive. This is done in order to encourage compliance with quarantine, as it allows the people near the infected person to be aware that that person is breaking quarantine.

Every day at 4 a.m. with an additional random delay of up to 30 minutes, the Smartphone contacts the Central Server and downloads its new Generator for the day. The Smartphone loads this new Generator, and will use it for all further IDs for the day. The system performs this action at 4 a.m. because it is a time when most people are asleep, causing the load on the server to be low. This and the additional random delay spreads the traffic to the server more evenly throughout time, increasing system performance.

Every 30 minutes with an additional random delay of up to 30 minutes, the Smartphone queries the Central Server to see if it has identified any new Generators as belonging to infected users (Infected Generators). The Central Server sends back any Infected Generators that it has identified since the Smartphone last contacted it so that the Smartphone can perform contact analysis using those Generators and its contact database. We chose 30 minutes as the interval to query the server because it represents an effective compromise between getting exposure results to users as quickly as possible while not overloading the Central Server with constant queries.

The algorithm for determining exposure uses the Generators belonging to infected people to generate all IDs representing such people, then counting the number of messages in the Contact table that have one of those IDs and was broadcasted from less than three meters away. If the total count represents an exposure time of over 20 minutes, the user is deemed exposed. Note that this algorithm treats all infected people as equivalent. 20 minutes of time with a single infected person counts the same as 10 minutes each with two different infected people. This is in line with current CDC guidelines for calculating exposure to COVID-19 (CDC, 2020, footnote 1).

The Smartphone also generates anonymized statistics on the collected data that are sent to the Central Server. The app generates four statistics: a count of how many unique infected people the user came in contact with, how many total messages the user received that were from infected users, how many total messages the user received that were from uninfected users, and the standard deviation of the number of messages from each infected user. These statistics were chosen because they preserve the user's privacy while also being useful to researchers. These statistics do not include data that can be used to positively identify who the user was in contact with except in degenerate cases such as only one person in the community being infected.

The Central Server

The Central Server is a well-equipped datacenter-class computer that the university owns. The university has given exclusive use to the computer to the contact tracing system. The Central Server stores general information on the students, such as their name, phone number, living situation, and the classes they are taking in a separate database. We explicitly consider only the information directly relevant to contact tracing, such as their current status (safe, isolation, quarantine) and the date any transitions between different statuses occurred, and their Generators. The Central Server also receives and stores the contact statistics the Smartphones generate. For similar reasons as the Smartphone, we again choose a relational database to store the tracing information. In addition to the previously discussed properties of reliability and durability, much work has gone into making relational database scale well in response to highly concurrent access. This is important for UniTrace as there will be many Smartphones accessing this data throughout the day, so having the database scale well is a priority. The database tables are shown below in Table 2.

Table Name	Contents
Users	User ID, Current Status (Safe, Isolating, Quarantine)
Transitions	User ID, Timestamp, Old Status, New Status, Cause (Exposure tracing, Living group infected, In-class infected, Positive test)
Generators	User ID, Date of Use, Generator State
Statistics	User ID, Date, Unique Infected Contact Count, Total Infected Message Count, Total Uninfected Message Count, Infected Message Count Standard Deviation

Table 2: Central Server Database

In order to support user privacy, the Central Server stores little more than what is required by university policy. In particular, the Central Server does not store enough data to deduce who exactly different users have been in contact with.

The Central Server does, however, store enough information to get a rough estimate of how much time the users have spent around other people. To encourage compliance with isolation and quarantine directives, this data can be used by the university to warn users who are breaking their quarantine or isolation.

The Central Server also interfaces with various medical testing facilities to ingest any relevant test results the facility has performed and integrate them into the system. The medical facility needs only to provide positive test results along with a timestamp of when the test was taken and the minimum amount of information necessary to uniquely identify a student. This is sufficient for the system to update any relevant user statuses. To support user privacy, the system does not permanently store any information from the test results, save for any transitions between user statuses as a result of the test.

When the Central Server receives a report of a positive test, it maps the test to a student, then changes their status to ‘quarantine.’ Current research suggests that people may be contagious up to 48 hours before a positive test (Harvard Health Publishing, 2020), so the system grabs the user’s current Generator, along with their Generators for the previous two days and puts them into the list of Infected Generators. Every time the Central Server generates a new set of Generators for every user, it will add the Generator of any user currently marked as ‘quarantine’ to the list of Infected Generators. This list of Infected Generators gets sent out to the Smartphones as they contact the Central Server throughout the day.

The Routers

The routers serve only their normal function of forwarding traffic along the network. We did not find any usages of this module that would help the system function while also preserving user privacy.

Events

To summarize the above, here is how UniTrace responds to the different events that can occur in the system:

New User joins UniTrace

To join UniTrace, users must download the contact tracing app and log into the app using their university login. After that, the central server generates a Generator, saves it, and sends it to the user's smartphone.

User passes by another app user

Every smartphone generates a BLE beacon once every ten seconds containing the current ID. When a user's phone receives a beacon from another app user, it notes the ID and time and saves this data locally.

User has a positive test

First, the testing center notifies the central server that one of its users had a positive test. Next the server looks through the database of Generators and finds all Generators associated with the infected user from the past two days. The smartphones query the server about every hour to get the new list of Infected Generators.

User's smartphone receives a list of infected Generators

The smartphone generates a list of IDs that each Generator can create. If IDs associated with one of the Infected Generators are seen every ten seconds for at least 20 minutes, the user is notified of an exposure event.

New Day

Every day, the server generates and distributes a new Generator for each user.

Conclusion

UniTrace will help universities track and reduce the spread of infectious disease. Our system preserves as much user privacy as possible, without sacrificing the contact tracing functionality. By checking for positive tests every half hour, UniTrace does not spend too much time checking and finding no updates when community infection rates are low, but still notifies people quickly when a positive test does occur. When test rates are high, the system can still handle checking every Infected Generator against its contacts, since a university has a relatively small number of people.

Works Cited

CDC. (2020, February 11). *Coronavirus disease 2019 (COVID-19)*. Centers for Disease Control and Prevention. Retrieved March 29, 2021, from https://www.cdc.gov/coronavirus/2019-ncov/hcp/guidance-risk-assesment-hcp.html

Harvard Health Publishing. (2020, March). *If you've been exposed to the coronavirus*. Harvard Health. Retrieved March 29, 2021, from https://www.health.harvard.edu/diseases-and-conditions/if-youve-been-exposed-to-the-coronavirus