Security & Privacy Analysis of Apple&Google's Contact Tracing

CS161: Computer Security

Ryan Lehmkuhl

August 10, 2020

Slides adapted from Raluca Ada Popa
Some content taken from: https://www.blog.google/documents/57/Overview_of_COVID-19_Contact_Tracing_Using_BLE.pdf

Contact Tracing

What is contact tracing?

- Identification of individuals who may have come into contact with an infected person

Why is it important?

- Notified individuals can do pre-emptive testing and quarantining

How is it traditionally done?

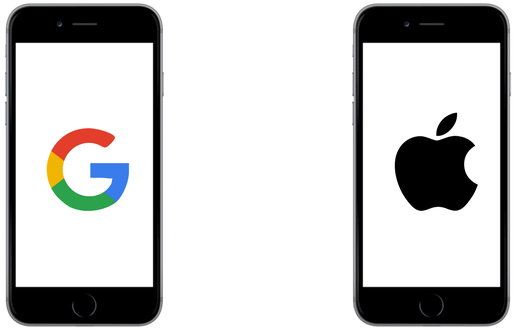
- Health officials will talk to infected individuals

Why reinvent the wheel?

- COVID-19 is highly contagious - these older techniques aren't scaling
- We need to automate tracing without violating people's privacy

Apple & Google’s contact tracing protocol

- The two companies teamed to create a decentralized contact tracing tool using which users can determine if they were exposed to COVID-19 with privacy and security considerations at its core
- Why the two companies in particular?



Uses Bluetooth technology

- Because COVID-19 can be transmitted through close proximity
- Bluetooth range: ~33 feet/10 meters
- If someone is in your Bluetooth range, there could have been a contact

Privacy and security are core to the algorithm

Why?

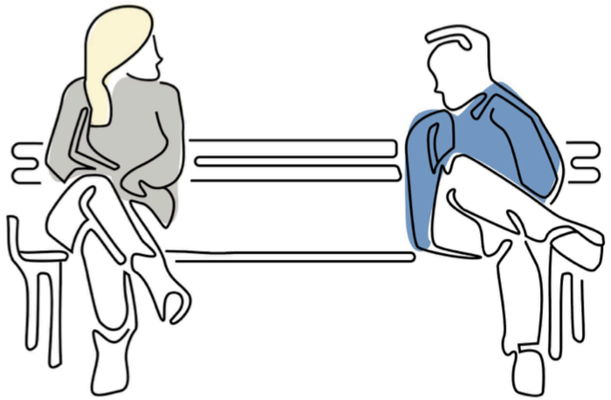
- Tracing involves private location data
- Revealing identities of infected individuals could lead to abuse
- Malicious users could try to tamper with the tracing

User consent

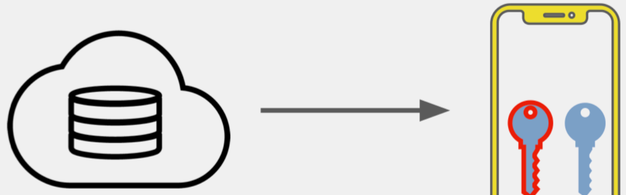
- Apple & Google provide an API that an official app can use
- Opt-in to install app
- Opt-in to declare if diagnosed with COVID-19

Workflow

Alice and Bob meet each other for the first time and have a 10-minute conversation.



Alice's phone periodically downloads the broadcast beacon keys of everyone who has tested positive for COVID-19 in her region. A match is found with the Bob's anonymous identifier beacons.



Anonymous identifier keys are downloaded periodically

A match is found

Alice sees a notification on her phone.

Alice's phone receives a notification with information about what to do next.

Additional information is provided by the health authority app or website

Google

The cryptographic protocol*

running on each user's phone

Every 24h period i:

- Generate Temporary Exposure Key $\mathbf{tek_i}$

  $\mathbf{tek_i} \leftarrow$ CRNG(16)

- Generate Rolling Proximity Identifier Key $\mathbf{RPIK_i}$

  $\mathbf{RPIK_i} \leftarrow$ HKDF($\mathbf{tek_i}$, "EN-RPIK", 16)

Every 10 minute epoch j:

- Generate a Rolling Proximity Identifier

  $\mathbf{RPI_{i,j}} \leftarrow$ AES($\mathbf{RPIK_i}$, "EN-RPI" || j)

- Transmit $\mathbf{RPI_{i,j}}$ via Bluetooth to all phones nearby

Receive:

- For every advertisement reception, store ($\mathbf{RPI_{i,j}}$, i) pairs locally.

If user is diagnosed:

- Release ($\mathbf{tek_i}$, i) of this user for the last some-number of days (ie. 14)

*Some details of the protocol have been omitted

Diagnosis server

- Aggregates all keys for the past N days
- Serves them to each user downloading periodically
- User identity and contact information is not uploaded to the server operator -> contact tracing is performed entirely locally

The cryptographic protocol*

running on each user's phone

Every 24h period i:

- Generate Temporary Exposure Key $\mathbf{tek_i}$

  $\mathbf{tek_i} \leftarrow$ CRNG(16)

- Generate Rolling Proximity Identifier Key $\mathbf{RPIK_i}$

$\mathbf{RPIK_i} \leftarrow$ HKDF($\mathbf{tek_i}$, "EN-RPIK", 16)

Every 10 minute epoch j:

- Generate a Rolling Proximity Identifier (RPI)

  $\mathbf{RPI_{i,j}} \leftarrow$ AES($\mathbf{RPIK_i}$, "EN-RPI" || j)

- Transmit $\mathbf{RPI_{i,j}}$ via Bluetooth to all phones nearby

Receive:

- For every advertisement reception, store ($\mathbf{RPI_{i,j}}$, i) pairs locally.

If user is diagnosed:

- Release ($\mathbf{tek_i}$, i) of this user for the last some-number of days (ie. 14)

Periodically:

- Download all new keys ($\mathbf{tek_i}$, i)
- Generate every related RPI and check against stored advertised pairs.
- If a match is found, you've been in contact with a COVID-19 patient.

*Some details of the protocol have been omitted

Security analysis

Privacy analysis

What sensitive information should we worry about in this app?

- Time-Location samples of each user
- Can identify who the user is and where-when they have been and with whom they came in contact

Privacy analysis

What private information do users see?

- Distinction between what a client app can see and what the user sees on the screen from an honest app
- For user Bob who declared COVID: Alice's client could figure out who the user was and where she met him. If a few users come together who were around Bob, they could reconstruct all the time-location path of Bob. Basically, you should assume no privacy guarantees if you declare you have COVID.
- For users who did not declare COVID, you could track the user for 10 minutes using the RPI, or more if more clients collude or if more contact

Privacy analysis

What private information does the server see?

- For users who declared COVID: their rolling identifiers. Put together with location data (e.g., from some users) it can identify the individuals.

- Less for non-diagnosed users: number of users, when they check for updates. Any information received from users colluding with server.

Privacy analysis

What other attacks could there be?

- Install recording devices in many places. Reconstruct identity and path of users who declared COVID
- DoS by broadcasting a huge number of RPIs
- Replay RPIs throughout different parts of the world
- Other ideas?

Consequences of no privacy for opt-in diagnosed users

Users might be afraid to declare they have COVID because:

- People might mistreat them (including violence cases)
- Someone who contracted from this user could hold a grudge forever

Integrity analysis

Can a malicious server affect the correctness of the tracing?

- Yes, entirely. This protocol trusts the server for integrity

What can malicious users do?

- Create false positives: upload fake “COVID” diagnosis and create panic; broadcast their RPI ids in many places in the world by replaying it there to create a lot of contact;
- Cannot prevent honest user with COVID to upload their own diagnosis unless the attacker can jam the network for that user or receiving users

In summary

- Contact tracing is crucial for controlling the spread of the virus
- Google and Apple's contact tracing protocol via Bluetooth aims to do this in a secure manner which protects user privacy
- Users without COVID-19 have some degree of privacy, those with COVID-19 have less
- Unfamiliarity with the technology has been the primary factor hampering adoption