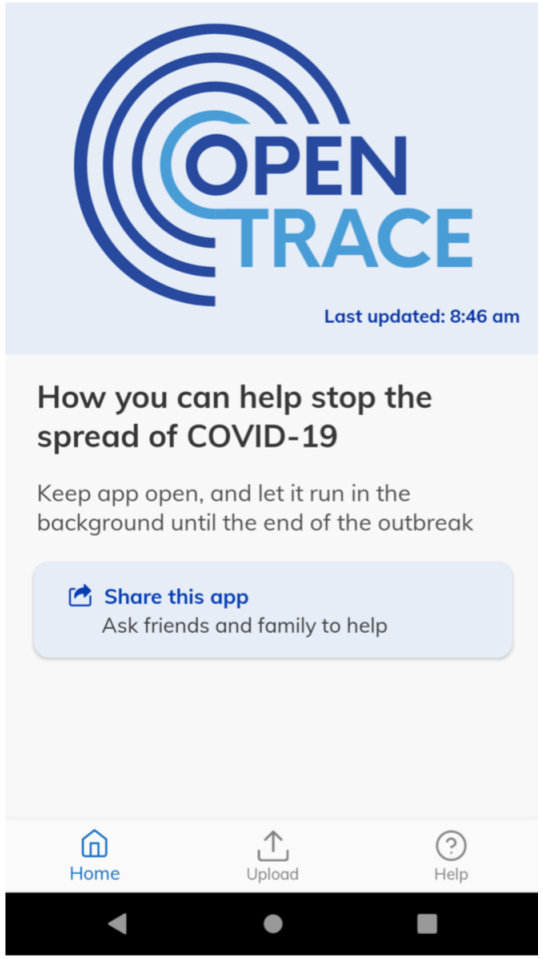
Coronavirus Contact Tracing App Privacy: What Data Is Shared By The Singapore OpenTrace App?

Doug Leith & Stephen Farrell,
Trinity College Dublin, Ireland
doug.leith@tcd.ie

https://www.scss.tcd.ie/Doug.Leith/pubs/opentrace_privacy.pdf

Coronavirus Contact Tracing Apps

- Of much interest at the moment, hope is they will help with easing current lockdown in various countries
- Use Bluetooth to infer contact events
- Singapore TraceTogether app seems to have been the first app using Bluetooth in this way to be widely deployed. Other countries have naturally been taking a close look at it.
- OpenTrace is open source release of TraceTogether



Privacy Concerns

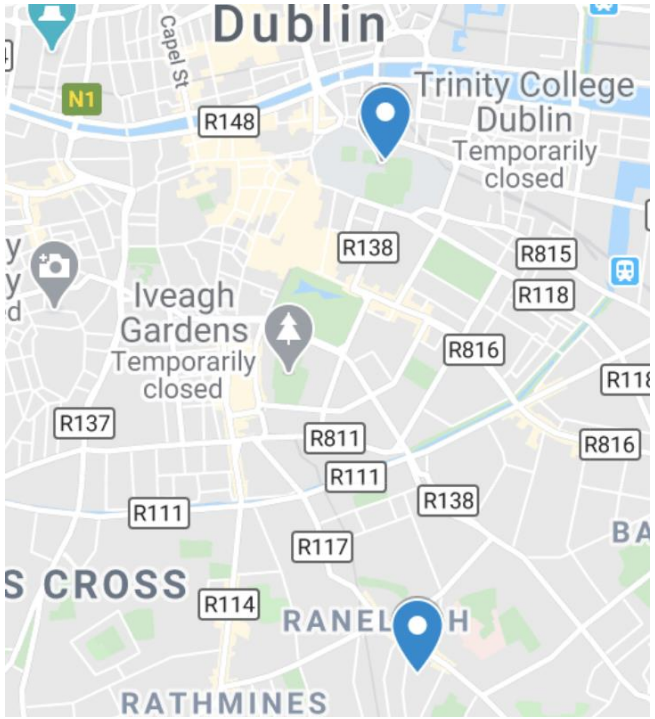
Privacy of these apps is attracting a **lot** of public interest

- Government sponsored apps that are being promoted for use by the whole population of a country
- Discussion mostly focusses on architectural issues (centralized vs decentralized etc)
- We argue its important also to look at the actual app implementation: its easy to have unexpected data release due to bugs, poor software choices etc

Privacy Threats

Its not just the data, we need to look at the meta-data

- Every message sent by app to a backend server includes an IP address
- Can IP address to location (geoIP services)
- If messages can be linked together then backend can construct a user location time history. We know that these are easy to deanonymize.
- Messages can be easily linked if they contain a persistent identifier e.g a device or app instance id



What We Did

- Downloaded OpenTrace, followed the instructions to setup the backend services that it needs, compiled the app and installed it on a rooted android phone.
- The backend traffic is of course encrypted. Certificate pinning is used.
- We used Frida to hook the java code and override the cert checks
- Then used Mitmproxy to intercept and decrypt the traffic sent by app to backend servers.

```
GET https://app-measurement.com/config/app/1%3A195668253963
%3Aandroid%3A0e1d84bec59ca7e66e160e
Parameters:
   app_instance_id: f67be0634d5102bcfe0352bc0bbeaded

\x07androidJ\x019R\x07Pixel 2Z\x05en-us'<j\
x0emanual_installr\x16io.bluetrace.opentrace\x82\x01\x191
.0.41-debug-b39f57f-F4D3\x88\x01\xa0\xac\x01\x90\x01\xf9\x8a
\x01\x9a\x01$1d2635f5-2af7-4fb3-86e...\xa0\x01\x00\xaa\x01
f67be0634d5102bcfe0352bc0bbeaded\xb0\x01\xda\x8f\xd6\xd8\xe4
\xe5\xa5\xf4\x8e\x01\xb8\x01\x03\xca\x01-1:195668253963:
android:0e1d84...\xd0\x01\x89\xc7\xdb\x92\x9b.\xe0\x01\x01\
xf2\x01\x16f4vnM2vqSLuOgcpB8FbDd_\xf8\x01)\x98\x02\x98\x9b\
xbe\xa5\xfa\xf8\xe8\x02\xa0\x02\x00\xe8\x02\xb2\xeb\x86\x0b\
xf0\x02\x00
```

What We Found

- OpenTrace uses Firebase as its backend. Means there are two parties handling data: Google and Health authority
- Google Analytics used to track user events e.g. when app is opened, closed etc. Steady stream of messages sent to backend servers tagged with id's.
- Uses Firebase Functions. A detailed log of every function call, per user, is kept
- User phone numbers are stored in Firebase Authentication service, which always runs on US servers
- A single long-term secret is used to reversibly encrypt phone numbers broadcast over Bluetooth. A single point of failure, breach would allow data publicly broadcast over Bluetooth to be decrypted to obtain phone number.
- Recommendations: Switch off analytics, don't use Firebase, use better crypto.
- https://www.scss.tcd.ie/Doug.Leith/pubs/opentrace_privacy.pdf