# Testing Apps for COVID-19 Tracking (TACT)

Stephen Farrell

stephen.farrell@cs.tcd.ie

Doug Leith

doug.leith@scss.tcd.ie

IRTF PEARG @ IETF108
July 2020

https://down.dsg.cs.tcd.ie/tact/

# Background/Overview

- So far during the COVID-19 pandemic, the Irish Government and Health Services Executive (HSE) are locally perceived to have done an overall great job managing the pandemic and already-stressed health care services
- A March 2020 paper (*) asserted that: “A mobile phone App can make contact tracing and notification instantaneous upon case confirmation.”
- Most people would like that to be true
- We were unsure if that was true or not, nor what privacy/security consequences might flow from population-scale uses of such Apps
- Trinity College Dublin (our employer) announced quick-turnaround funding for projects related to the pandemic (mostly aimed at medics)
- We applied for, and got funding for, Testing Apps for COVID-19 Tracing (TACT)

(*) Ferretti, Luca, et al. "Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing." Science 368.6491 (2020).

# Caveats

- This is a **very** fast-moving area
  - E.g. Just to prove that: Google/Apple published some of their internal code on July 23rd as I was writing these slides;-) – We've not analysed that yet, but it's a good thing
    - https://github.com/google/exposure-notifications-internals
    - https://developer.apple.com/exposure-notification/
- Many decisions, esp. earlier ones, had to be made quickly in the face of significant uncertainty
- Lots has changed in the last couple of months, and more change seems likely to us
- We also believe everyone with whom we've interacted on this so far is trying to do good
- But: it's also important to improve/fix things that need fixing

# Overview

- The Google/Apple Exposure Notification (GAEN) system
- A replay attack
- Measuring deployments
- Is BLE proximity detection effective?
- What traffic do GAEN Apps send? (on Android)
- Conclusion

# The GAEN System (1)

- Details at: https://www.google.com/covid19/exposurenotifications/
- GAEN App uses GAEN API implementation
- App handles interaction between handset and Public Health Authority servers
- API implementation handles key storage, BLE beacons and beacon/key (TEK) matching
- Handsets generate a Temporary Exposure Key (TEK) every day
- BLE beacons are sent @~4Hz and contain a Rolling Proximity Identifier (RPI) value that changes whenever BLE MAC address changes (about every 10 minutes)
  - $RPIK_i$ := HKDF($TEK_i$,NULL,UTF8("EN-RPIK"),16)
  - $RPI_{i,j}$ := AES128($RPIK_i$,time-in-10-min-chunks)
  - Beacons also include encrypted TxPower value (unauthenticated encryption but not that bad here)
- Handsets listen for beacons for about 4s every 4 mins and record the beacon value and Received Signal Strength Indicator (RSSI) for 14 days
- If Alice tests positive, she causes the App to upload the set of TEKs she used for the last (up to) 14 days to her Public Health Authority server
- Bob's handset downloads TEKs from his Public Health Authority server, perhaps every 2 hours, and checks if any TEK matches any stored RPI

# The GAEN System (2)

- Stated goal of these Apps is to detect if two handsets were within 2m for more than 15 minutes, based essentially on matching RPIs to TEKs and the associated RSSI and TxPower values
  - Spoiler: I don't believe that can be reliably achieved
- "Attenuation" is the measurement used, defined as TxPower-RSSI
  - TxPower might be -27dB, RSSI might be -80dB so attenuation then would be 53dB
  - Measurements also need to be amortised over matching beacons seen and maybe (not specified) with some outliers thrown away
- GAEN API implementation accepts thresholds from App code and returns attenuationDuration values for the "above", "between" and "below" ranges
  - E.g. App might input "[55,62]" and, if there's an RPI/TEK match, get back "[10,3,11]" meaning handsets were "closer" than "55dB attenuation" for 10 minutes, in between 55db and 62dB for 3 minutes and "further away than" 62dB for 11 minutes
- App then decides if that output implies a notification is needed
  - If notified, user is usually guided to isolate, go get tested, etc.

# The GAEN System (3)

- Governance: Google and Apple are in control
- (IMO) Reasonable justifications for that:
  - There are ~200 countries, don't want 200 schemes
  - Google/Apple mobile OS duopoly, and their knowledge of handset internals, means you couldn't credibly tackle this problem without them
  - Don't want: fake tracing Apps, Apps draining battery messing with BLE or Apps using worse crypto than GAEN spec
- Upshot however is that OS updates can directly affect notifications without any visibility to Public Health Authorities (e.g. calibration adjustments)
  - Spoiler: revisiting governance may be a good idea

# The GAEN System (4)

- Some interesting questions for these Apps might be:
  - How many people who would not have been found via manual contract tracing get notified?
  - How many people are notified sooner than would be the case with manual contact tracing?
  - How many notified people turn out to test positive for COVID-19, vs. the averages for testing at that time/in that locale?
  - What are the true/false positive/negative rates for notifications (where “true” == “within 2m for >15 mins” or whatever is the goal)
- So far, we’ve not seen the overall contact tracing systems (manual+App-based, together) being setup so as to be able to answer questions like those above. It’d be good if they were.
- How many downloads or the proportion of the population who’ve installed isn’t really a good metric
  - “You need 60% of the population” is not true – that refers to an unrealistic model in Ferretti et al where the only contact tracing is via such Apps

# Replay Attack

- There’s an obvious replay attack: collect beacons (likely from someone who’s positive) and re-tx/spread those to others elsewhere
  - https://down.dsg.cs.tcd.ie/tact/replay.pdf
- We calculate an (under)estimate for the amplification factor for such an attack
  - With conservative early-May Irish figures, “collector” at COVID testing station and “spreader” at hospital emergency department, each true-positive case could produce 4 or more false positive notifications
- Attack is obvious, hasn’t been mitigated and hasn’t (yet) happened (AFAIK)

# Measuring Deployments

- The set of TEKs that Bob's phone downloads are public, so we can count those
- We're currently measuring those for the Irish, Italian, German, Swiss, Polish, Danish, Austrian and Latvian apps (and for the Spanish one being trialled)
  - https://down.dsg.cs.tcd.ie/tact/tek-counts/
  - Recall that you upload 14 TEKs, and the health authority might publish some or all of those, so if we see 1 new TEK for Ireland today, there'll (maybe) be 1 new TEK added to the count for each of the last 14 days – we do have hourly snapshots so can reconstruct when TEKs are being added, but that's not yet done
- Still too early to draw any real conclusions
  - Server implementations do some odd things (adding fake TEKs)
  - Some TEK counts do seem small relative to number of cases declared in that country
- Note: these measurements can't tell us if these Apps "work" but might tell us they don't work (in some places)

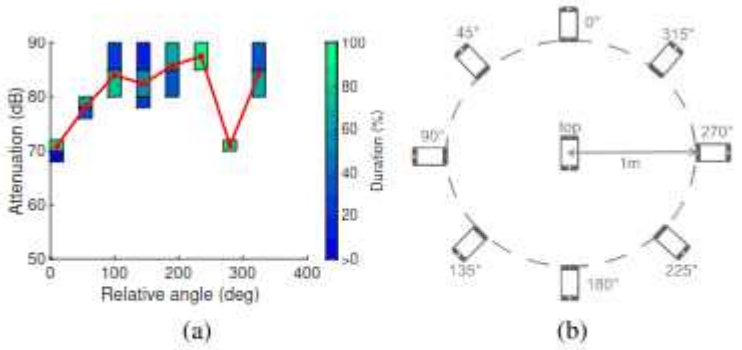
# Does BLE Proximity Work?

- Pairwise tests with different handset types at 1m for 30+ minutes still produce false negatives if you assume some “noise” due to orientation that affects attenuation
  - https://down.dsg.cs.tcd.ie/tact/posorient.pdf
- On June 13th Google shipped an update that added new calibration adjustments to attenuation calculations. We had tested before that, so we re-did the same tests.
- Percentage false negative seen for various Country configurations:

| | Late June | | | | Early-Mid June | | |
|---|---|---|---|---|---|---|---|
| Country | -10dB FN% | 0dB FN% | 10db FN% | Country | -10dB FN% | 0dB FN% | 10db FN% |
| Austria | 0 | 0 | 21 | Austria | 9 | 36 | 82 |
| Denmark | 0 | 0 | 21 | Denmark | 9 | 36 | 82 |
| Germany | 0 | 0 | 21 | Germany | 9 | 36 | 82 |
| Ireland | 0 | 0 | 27 | Ireland | 9 | 42 | 82 |
| Italy | 0 | 0 | 0 | Italy | 0 | 18 | 55 |
| Poland | 0 | 6 | 52 | Poland | 21 | 67 | 85 |
| Latvia | 0 | 18 | 55 | Latvia | 21 | 79 | 85 |
| Switzerland | 0 | 0 | 33 | Switzerland | 18 | 55 | 82 |
| Overall | 0 | 3 | 29 | Overall | 12 | 46 | 79 |

# “Noise”

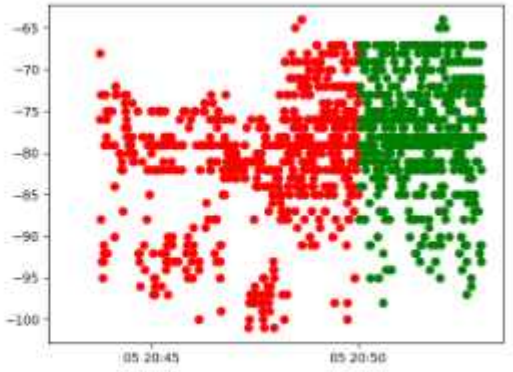
- Handsets package antennae in different ways and so orientation can change attenuation by itself (as can other things)
- We’re not sure how to model this but it seems to be able to affect RSSI (and hence attenuation) by 10-20dB




(a) (b)

Fig. 4: (a) Measured attenuation duration vs the angle between two handsets placed 1m apart. (b) Schematic showing orientation to which each angle corresponds.

(a) Setup

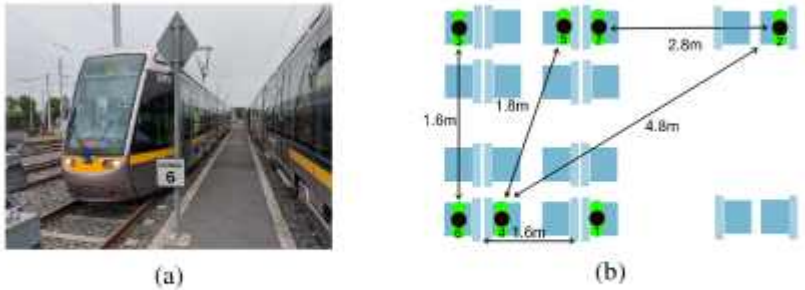


(b) RSSI

Cat video! https://down.dsg.cs.tcd.ie/tact/changes.mov

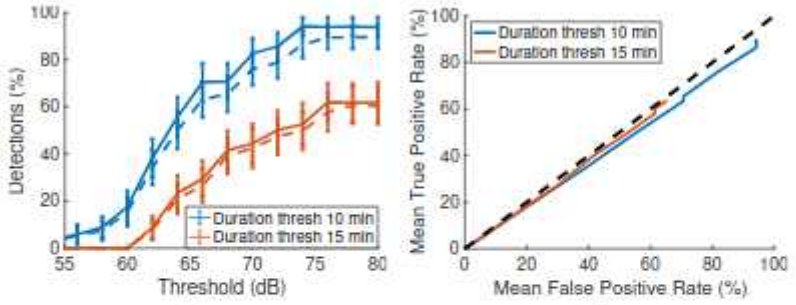
# Real-World Scenario Testing

- We did tests of real-world scenarios to see how those affect RSSI, attenuation etc.
  - Walking, cycling, sitting around a table, on a park bench, between cars in parallel
- Two are noteworthy:
  - On a commuter bus https://www.scss.tcd.ie/Doug.Leith/pubs/bus.pdf
  - On a tram https://www.scss.tcd.ie/Doug.Leith/pubs/luas.pdf
- Those seem like scenarios where these Apps, if they work, could help with contacts that would otherwise be missed but the metallic surrounds affected BLE distance estimation badly




(a) (b)

Fig. 1: (a) Tram on which measurements were collected. (b) Relative positions of participants during tests.




(a) (b)

Fig. 8: Exposure notification true and false positive rates when a simple threshold strategy is applied to the GAEN tram dataset. (a) True and false positive rates vs attenuation level and duration thresholds, solid lines indicate true positive rates and dashed lines the corresponding false negative rates. (b) ROC plot corresponding to mean rates in (a), dashed line indicates 45° line.

# What traffic is sent? (on Android)

- The Android implementation of the GAEN API is a part of Google Play Services. If you disable Google Play Services these Apps won't work.
- We mitm'd and unpinned a set of GAEN Apps and captured traffic traces of App traffic and traffic from Google Play Services
  - https://www.scss.tcd.ie/Doug.Leith/pubs/contact_tracing_app_traffic.pdf
- Overall the Apps seem well-behaved but Google Play Services shares long term identifiers with Google and also frequently connects in a way that would allow IP-based location tracking

# Example: Irish CovidTracker App

- For all apps, we only tested onboarding and TEK downloads – we didn't touch anything that the client might e.g. do after a positive test as that could interfere with a running service
- Irish App has some issues:
  - An unnecessary "supercookie" – A JWT authToken used as a bearer token to download TEKs (no other services had such an individual cookie) – the HSE's Data Protection Impact Assessment (DPIA) states that they don't log that kind of information but it'd be better to not have it in the protocol
  - The App allows users to opt-in to sending "metrics" that mix devops (whether App opened that day) and medical information (how many times notified) – we recommend separating those into different security contexts
  - There's some path-not-taken code that makes a call to Google Firebase
- Most apps were open-source, had DPIAs, did certificate pinning and were pretty clean in terms of data-sent, Irish App would be "mid-table" if this were a league

# Google Play Services

- We turned off everything we could, including Google Play Services “Usage & Diagnostics” and tried to get to a minimal configuration where a GAEN App can run (reminder: that requires Google Play Services to be running)
- Roughly every 6 hours, Google Play Services connects to a “/checkin” API and sends back information including the handset IMEI, phone number, SIM serial number, WiFi MAC address and the email address associated with the handset
- Roughly every 20 minutes, Google Play Services connects to a “/p/log/batch” API including a cookie that is present in the “/checkin” HTTP request, thereby linking many long-term persistent hard-to-change identifiers with the IP address that can be geo-located
- The above messages contain additional telemetry – how much depends on settings and what other Apps are running (but is opaque); there was some variation in the frequency of connecting to the “/p/log/batch” endpoint, depending on settings.
- There is no published DPIA for the Android GAEN API implementation nor of Google Play Services (that we know about).
- Google Play Services is closed-source.
- That all seems hugely invasive to us and is required if you want to use a GAEN App on Android.

# Conclusions

- These Apps are being deployed, governments are encouraging adoption for entire populations
- Replay attacks (and perhaps others) exist and have yet to be mitigated
- BLE-based proximity detection may not work as well as claimed
  - We do expect that to improve over time, but are unsure if it will ever be reliable
  - The “<2m && >15 mins” criterion may be asking the wrong question – we find it hard to see that that can be reliably determined, we don’t know if some other (epidemiologically) useful criterion could be reliably determined
- The Android implementation of the GAEN API comes with serious privacy problems
  - If you don’t care about Google tracking you, then you have no problem
  - If you don’t want to install a GAEN App, then you have no new problem
  - If you care about Google tracking you and want to run a GAEN App, you have a new problem
- We have no information about the internals of the Apple implementation
- The GAEN system still requires more public documentation, perhaps needs a new “quiet mode” to justify population-wide acceptability and it may be wise to revisit the governance setup for the overall GAEN system

# Resources

- All details are available at: https://down.dsg.cs.tcd.ie/tact/
- All but one of the documents there are tech-reports, not (yet) peer-reviewed
- We have published data sets for the bus, tram and pairwise data sets and publish the daily TEK counts
- We have published code for the TEK survey and a modified version of the Google exemplar GAEN App that was used in tests
- We have an (unpublished, it'd help with the replay attack) App that supports other tests by doing the transmitting GAEN functions without using the GAEN API but that interops with the Android and Apple implementations – happy to make available to other researchers

# Thanks

Offline questions welcome too
stephen.farrell@cs.tcd.ie

These slides:
https://down.dsg.cs.tcd.ie/tact/