Hansel and Gretel and the Virus - Privacy Conscious Contact Tracing*

Jaap-Henk Hoepman

In an effort to monitor or control the spread of COVID-19 (aka the Corona virus), countries have turned to invasive forms of surveillance based on the location data that mobile telephone operators collect of their users. This is not only happening in China or South Korea, but also in Israel and even Germany, Austria and Italy. The details differ per country.

In this blog post I want to describe the current ways in which location data is being used to fight the spread of COVID-19, discuss why this worries me for the future, and describe some technological options for more controlled, privacy conscious tracking of infected people. (I hesitate to write privacy friendly in this context.)

1 What is going on?

In China and South Korea, the itineraries of infected people are published online so other people can determine whether they have been in the same area at the same time. Online services have appeared to ease the process: "By entering the date of their journey, together with flight or train numbers, users can find out if they were traveling with someone who has been infected with the virus." ->. Israel is also tracking the mobile phones of suspected coronavirus cases. According to this piece (in Dutch) the Israeli security service Shin Beth is even tracking people that were in the vicinity of infected people up to fourteen days before the diagnosis was made, and informing them about this by SMS. (*Confirmation of this through English sources would be appreciated. Also, I believe I read somewhere that a similar automatic system for tracking people in the vicinity of infected people was used in China, Hong-Kong, Taiwan or South-Korea. Again I could not find a proper source. If you know any sources please add them to the comments section below.*). Germany, Austria and Italy have asked mobile phone operators to share anonymous, aggregated data with health authorities to help monitor lockdown compliance. The

* Wed Mar 25 10:54:14 2020 +0100 / 2654ca0 / hansel-and-gretel-and-the-virus.md

European Commission is also advocating such a scheme for Europe at large.

The European Data Protection Board has issued a statement on the processing of personal data in the context of the COVID-19 outbreak. Regarding the processing of telecom data, such as location data, it states that

> *national laws implementing the ePrivacy Directive must also be respected. In principle, location data can only be used by the operator when made anonymous or with the consent of individuals. However, Art. 15 of the ePrivacy Directive enables Member States to introduce legislative measures to safeguard public security. Such exceptional legislation is only possible if it constitutes a necessary, appropriate and proportionate measure within a democratic society. These measures must be in accordance with the Charter of Fundamental Rights and the European Convention for the Protection of Human Rights and Fundamental Freedoms. Moreover, it is subject to the judicial control of the European Court of Justice and the European Court of Human Rights. In case of an emergency situation, it should also be strictly limited to the duration of the emergency at hand.*

2 My concern

We are witnessing an unprecedented invasion of our fundamental human rights in an effort to fend off a total collapse of our health services and the resulting death of perhaps millions of people (that could otherwise have been partially prevented). I do not wish to start a discussion on the proportionality of the measures that have been taken at the moment (although I think we definitely should be having such a discussion at a later stage). All I want to say right now is that I have the utmost respect for everybody responsible for navigating their 'ship' in such times of crisis where the slightest mistake can have disastrous consequences, while at the same time there is little reliable information available to base rational decisions on.

My concern is what will happen once we have overcome this crisis. Or perhaps what will be hacked together quickly to overcome this crisis in the first place.

Current evidence suggests that the massive surveillance system that China already had in place has helped it to curtail the spread of COVID-

19. Italy now (March 23) has more Corona related deaths than China, who has in fact managed to all but stop the spread of the virus. This increasing pressure on other, Western, governments to roll out similar surveillance systems to prevent a new outbreak of COVID-19 this fall, or another virus some years down the road. Suppose we decide (after a proper debate I alluded to above) that such a system allowing governments to track the whereabouts of infected citizens and their contacts is proportional in times of a pandemic. Then the question is whether we can somehow prevent that such a system is used for other, more nefarious, purposes by oppressive regimes.

It is with this concern in mind that I will outline some possible approaches to the problem of tracking of people that have been infected by a virus, and in particular automated ways to trace all people they have been in contact with, in a more privacy conscious way. I would like to stress that I am in no way advocating that tracking infected people in this way is the right thing to do. In fact I believe this is a road to hell paved with good intentions. But I do feel that for any meaningful debate on this issue it is necessary to have a proper understanding of what is and what isn't possible from a technological perspective. And I do fear that if we do not have any privacy conscious approaches ready soon, an all-out mass location tracking system will be imposed upon us instead.

3 Location based tracking

Most location tracking systems deployed at the moment to track people infected by the COVID virus rely on the location data that mobile phone operators collect about their users. In order to deliver text messages or route incoming calls, mobile phone operators need to keep track of the *current* location of all their users. This location typically corresponds to the area covered by the base station that the phone is currently connected to. (Although it is certainly possible that in certain countries this location is recorded in much more detail with the help of GPS data provided by the phone itself, or using more detailed information available to the mobile network itself.) The precision of the location thus depends on the area. Depending on the type of network, and whether the user is in a densely populated city or in the countryside, the precision of the location can range from several hundred meters to tens of kilometres. In other words, this location data is less accurate than GPS, which has an precision of roughly 5 meters (under ideal circumstances).

Mobile phone operators do not only keep track of the current location of their users, but also record *a history* of their location (i.e. the base station connected to) in so called Charging Data Records (CDRs) that are created whenever people make or receive calls or messages. The amount of detail in such CDRs differs among operators. This data is retained for as long as required for billing and fraud prevention purposes, which easily extends to several months. In Europe this data used to be retained for six to twelve months under previous data retention laws. Dutch telco's store aggregate data (not including location) for taxation purposes for seven years.

Note that apps that offer location based services on users smartphones using GPS have a much more detailed picture of the actual user location, and often maintain a history of previously visited locations (e.g. Google's location history collected when you use Google maps while being signed in to your account).

4 The problem

In technical terms, the problem we aim to solve here is called *contact tracing*: to automatically find the people that recently have been in close physical proximity with a person that is diagnosed with a COVID-19 infection. Using mere cell phone location data is not accurate enough to do this: GPS data or other location data with similar accuracy (e.g. obtained through triangulation or based on proximity to Bluetooth or WiFi receivers) is required.

In fact the British National Health Service (NHS) recently started developing a very basic app for exactly this purpose, emulating a system deployed in China. People voluntarily install an app on their phone that tracks their GPS location which is recorded in a central contact tracing database (presumably maintained by the NHS). As soon as someone that installed the app gets diagnosed with the infection, this is reported to this central database, and all other people that also installed the app and have been in close proximity to this person the last two weeks automatically get a warning and instructions on what to do next: whether to self-isolate, adhere to social distancing, or to get tested. Details are described in this preprint "Sustainable containment of COVID-19 using smartphones in China: Scientific and ethical underpinnings for implementation of similar approaches in other settings", which also argues why this method of 'herd protection' (instead 'herd immunity') works.

This system introduces obvious risks of mass surveillance. Here we investigate more privacy friendly solutions to this contact tracing problem.

5 A related problem: locating lost devices

Our solutions are inspired by solutions to a related problem, namely that of locating lost devices. In particular, we use ideas present in recent improvements to Apple's 'find-my-iPhone' feature that allows devices without a GPS sensor (like certain iPads or iWatches) to be located as well. Their system (supposedly) works as follows.

Let L be the lost device. Before getting lost, this device was paired with a trusted partner P (typically another device owned by the same person). People are assumed to immediately pair new devices with devices they already own: without this earlier pairing, lost devices cannot be found. Device L has a secret identifier id_L known to P. Paired device P has a public key K_P known to L. L uses its secret identifier id_L to generate a random pseudonym rid_L, for example by hashing the identifier and the current time t using a cryptographic hash function h, that is $rid_L = h(id_L, t)$. L broadcasts this random pseudonym every once in a while over its local Bluetooth and/or WiFi interface, in the hope that some helper H (e.g. anybody else's iPhone that happens to be nearby) receives this signal. Randomising the identifier is necessary to prevent anybody from singling out and tracking L through its static identifier id_L. (This also assumes the MAC address of the Bluetooth or WiFi interface is properly randomised as well.)

This helper H is supposed to know its current location (because it does have a GPS receiver) and is supposed to send its location and the pseudonym of the device it discovered to Apple's iCloud servers. The trusted partner P trying to locate L will query these servers with the recently used pseudonyms for L using its knowledge of id_L that allows him to compute $rid_L = h(id_L, t)$ for any t in the recent past. If a tuple for any of these queries exists, the locations contained in it will be reported back as a result, allowing P to locate L.

In order to protect the location privacy of the helper H, the location it reports to Apple's iCloud needs to be encrypted (otherwise Apple would know the location of all helpers), and it needs to be encrypted against P's public key so P can actually decrypt it. The only way for H to know this key, is if the lost device L broadcasts it alongside its randomised pseudonym. But as P's public key is static, this would serve

as an ideal static identifier to track L, thwarting the use of the randomised pseudonym. This can only work if L can also randomise P's public key K_P before broadcasting it. Luckily, encryption schemes exist that allow the public key to be randomised while ensuring that the same unchanged private key is the only key that can be used to decrypt a message.

ElGamal is one such public key encryption scheme (-> and ->). Recall that ElGamal works in a cyclic group G with generator g. In ElGamal, public key $K = g^k$ belongs to a private key k. Encryption of a message m against a public key (K, g) (where we add the generator g to the description of the public key) is achieved by selecting a random r and computing $E_K(m) = (m \cdot K^r, g^r)$. Decrypting this ciphertext (c, d) using private key k is done by computing $c/(d^k)$ (using the property that $(g^r)^k = (g^k)^r = K^r$).

In this setting, the public key (K, g) can be randomised using any random factor s as $(K', g') = (K^s, g^s)$. Encryption of m using this new randomised key yields $E'_K(m) = (c', d') = (m \cdot K'^r, g'^r)$. Decrypting this with the original k s.t. $K = g^k$ using the formula c'/d'^k yields $m \cdot K'^r/(g'^r)^k$ which still returns $m \cdot K^{sr}/g^{srk} = m$ as required.

6 A privacy conscious protocol

Inspired by Apple's protocol, we could create a system where devices leave a trail of breadcrumbs (like Hansel and Gretel in their fairy tale) consisting of their location and identity. The system can be made privacy conscious by ensuring that this trail is not maintained in any central database (like the NHS proposal outlined above does), and making sure the breadcrumbs disappear after some time (unlike the original fairy tale, the disappearing breadcrumbs are not a bug but a feature ;-).

The idea is to let a phone collect the identity of any other phone it encounters in its vicinity over time. Like the Apple protocol, it uses the WiFi or Bluetooth short range local network for detecting other phones. Whenever it does, it records the current location. (The other phone, also seeing this phone, does the same.) This log is kept for any encounters within some predetermined and hard coded time limit L, like two weeks. Older entries are automatically (irrevocably) purged by the phone collecting them.

If a person is diagnosed to be infected with a dangerous virus, the log in his phone can be extracted. For this the phone needs to be in physical

possession of the competent health authorities. As the phone purges old encounters, the health authorities can never track movements older than the time limit L. But all recorded recent entries allow the health authorities to immediately identify all people that were at one time within the time limit L in the vicinity of this patient.

There are two requirements that must be met in order to make such a system privacy conscious. First, no phone should be able to learn the identity of any other phone in its vicinity. In other words, before broadcasting its identity id_B a phone needs to encrypt it against the public key of the health authorities K_A (using a semantically secure encryption scheme to ensure that two encryptions of the same identity cannot be linked). But to prevent the authorities from snooping on the communications between phones to get access to the broadcast identities directly, these encrypted identities need to be further encrypted against the public key K_C of the phone collecting them. As in Apple's find-my-iPhone case, broadcasting this key would allow the collecting phone to be tracked, so this key needs to be randomised.

The full protocol now looks like this for phones B and C meeting each other.

- C randomises its public key K_C to compute K'_C.
- C broadcasts this randomised public key K'_C over the WiFi or Bluetooth network.
- B encrypts its identity id_B and its current location loc_B first using the public key K_A (of the authorities) and then using the randomised public key K'_C it just received.
- B sends the result $E_{K'_C}(E_{K_A}(id_B, loc_B))$ back to C over the same WiFi or Bluetooth network.
- C decrypts this using its private key k_C and stores $E_{K_A}(id_B, loc_B)$ in its log together with its own current location.

The same protocol runs simultaneously with the roles of B and C reversed (and for any other phone that also happens to be in the vicinity). The log of C can be accessed and identities decrypted by the health authorities using their private key k_A.

Storage requirements are moderate: even when running this protocol every minute, and assuming on average 20 other phones are in the vicinity, the total number of records for a two week interval L equals $60 \cdot 24 \cdot 20 \cdot 14 = 403.200$. If each record requires 1 kB storage, the total storage demands are 403 MB.

The main problem with this protocol is the fact that two devices need to complete a full message exchange (send randomised key, wait to receive encrypted identity and location) over the WiFi or Bluetooth network. This may not be realistic or reliable.

7 Using central storage, in a privacy conscious way

This can be avoided if we do rely on centralised collection of the bread-crumbs that contain the identity and location of a phone. This only requires a minor modification to the above protocol: the first steps are essentially the same as before, but instead of sending its response back to the other phone C, the response (with encrypted location and identity) is sent to a central server maintained by the health authorities. There are some subtle differences however.

The full protocol now looks like this for phones B and C meeting each other.

- For every one-minute time slot, C creates a fresh private key k'_C and corresponding public key K'_C.
- C also logs its own location at this time slot (purging records older than the time limit L) together with the private key k'_C and public key K'_C just generated.
- C broadcasts this fresh public key K'_C over the WiFi or Bluetooth network.
- B encrypts its identity id_B and its current location loc_B first using the public key K_A (of the authorities) and then using the public key K'_C it just received.
- B sends the result $E_{K'_C}(E_{K_A}(id_B, loc_B))$ together with the public key K'_C to the authorities A using the cellular network.

Tracing the contacts of a person found to be infected is slightly more contrived now. If phone C in the protocol above belongs to such a person, the health authorities first extract its own location history log from this phone. This location history provides the authorities with all key pairs k'_C, K'_C used over the past period L to record the location of nearby phones. For every K'_C in this history, the authorities look up matching records in the database (there may be several when more than one phone was in the vicinity of C in that particular time slot). The database returns $E_{K'_C}(E_{K_A}(id_B, loc_B))$. Using the private key k'_C also present in C's log, the authorities can remove the first layer of encryption, and subsequently use their own private key k_A to decrypt the record and obtain the identity

id_B and the exact location loc_B of the phone B that was close to C at some point in time.

In the protocol above we let C generate fresh key pairs, to ensure that the private key is no longer available after the time limit L (because its record in the log is automatically destroyed by the phone after that time). To reduce the storage requirements for the local phone log, a hybrid approach using randomised ElGamal keys could be used instead. For example C could generate a truly fresh ElGamal key pair each day (recording the private once each day in the local logs), and derive randomised public keys K'_C every minute for use in the above protocol (recording all of those in the local log). This ensures that encrypted location information of other phones is guaranteed to be unencrypted one day after the time limit L.

Using one fixed private key k_C forever is also possible in principle, as the corresponding randomised public key K'_C necessary to find a matching record in the central database is always destroyed after time limit T. But this does not prevent the authorities to go for a massive phishing expedition trying to decrypt *all* records in the database with a private key k_C they just recovered.

8 Hiding further away devices

The above protocols would be even more privacy conscious if the amount of work that the health authorities need to invest in order to compile a list of all devices that once where in the vicinity of someone's phone would depend on the distance of these devices to the phone in question: the further away, the less relevant their identity is, and therefore the more effort the health authorities need to invest in order to still recover the identities of such further away devices.

When phones collect location data of nearby phones themselves (as in the first protocol), we can envision a data release protocol where the phone responds to requests from the health authorities with a delay that increases exponentially with the distance between the two phones in particular record in the phone history log. (This would require the phone to know the location of the other phone in the record, or at least an estimate of the distance. The first protocol above would have to be modified to provide this additional information.)

When phones send location records to a central server directly, as in the second protocol, this is a bit more difficult to achieve. One heuristic

would be to use an additional layer of encryption using a symmetric cipher using a randomly selected key where the number of known (0) bits of the key depends on the distance and an estimate of the current key search capabilities of the health authorities. With every linear increase in the distance then the difficulty decrypting the record raises exponentially. Note that even if such an estimate of the current key search capabilities of the health authorities is accurate *now*, it does little to prevent the authorities from decrypting all data easily through brute force some years later, when their key search capabilities has increased by several magnitudes due to Moore's Law.

An alternative approach based on the second protocol would work as follows. Instead of releasing the full location history containing all private keys k'_C immediately, phone C only releases the public keys K'_C to allow the authorities to collect all matching records in the central database. These records are then sent back to phone C with the request to decrypt them using the (privately kept) copy of k'_C. The phone can decrypt each record internally immediately, compute the local distance (which would require small modification in the protocol that ensures that only the identity and not the location is encrypted with the health authorities key K_A), and delay the release of the result to the health authorities depending on this distance.

9 Discussion

In the above protocols, the health authorities learn nothing without the cooperation of and physical access to the phones that collect the information. This relies on two assumption though.

1. Authorities cannot or do not install beacons pretending to be ordinary phones, thus allowing them to surveil all phones entering and leaving a particular location. This is difficult to prevent (authorities could simply buy legitimate phones and put them all over the place), although some level of protection is possible when phones only broadcast their location and identity against keys of other phones that are known to be really very close (assuming this can be reliably be measured using signal strength, which is tricky, or using general distance bounding protocols; further investigation is necessary here).
2. Phones really protect the location data collected in this way and *only* release that data when in physical possession of the health authorities, and only after clear authorisation by the user. By requiring

physical access, e.g. ensuring that the location history is *only* released over the USB or Lightning port, large scale surveillance based on remote access is prevented.

This is a draft paper describing work in progress. Your comments are most appreciated.