Privacy for Location Data in Safe Paths

Privacy is one of our core values, and we aim to provide as much privacy as possible to users of the app.

We believe that it is possible to deliver location-based exposure notifications to users with an absolutely minimal privacy impact. This paper explains the various measures that we are taking, and plan to take in future, to keep that privacy risk to an absolute minimum.

Privacy Goals

There are three key constituencies whose privacy we wish to protect.

Undiagnosed Users

Undiagnosed users are our top priority from a privacy point of view. The reasons for prioritizing this group are:

- They are our largest constituency of users overall
- For efficacy of the solution, we are dependent on widespread adoption by this group.

To provide the strongest possible privacy for these users:

- Location data for undiagnosed users never leaves their device (i.e. this is a “decentralized” solution).
- It is also encrypted at rest.
- It is deleted after 28 days
- The user can turn location tracking on & off whenever they like (either using the in-app setting, or their device settings)
- Data is deleted when a user uninstalls the App.

We believe that this keeps the privacy risks associated with undiagnosed users to the absolute lowest level possible.

Diagnosed Users

Diagnosed users are our next priority from a privacy point of view. In order to permit location-based exposure notifications, and for other Health Authority analysis purposes, it is necessary for these users’ location data to leave their device, but we aim to do this in the most privacy-preserving way possible.

- Sharing location data is always a choice on the part of a diagnosed user, with clear explanations provided about the purposes for which the data will be used.
- Location data can only be shared with authenticated Health Authorities that have been approved by Path Check.
- Location data is encrypted in transit, and at rest in the Health Authority’s database.
- Prior to storage in the Health Authority databases, a Contact Tracer will review all provided data with the diagnosed user, and redact any information that either of them believes could reveal their identity.
- Data is only stored in a databased owned and managed by the patient’s own Health Authority. There is no centralized store of location data.
- The patient can also request that any other data point recorded is removed, for any reason they may have.
- Data is only committed to the Health Authority database once the diagnosed user has given final consent regarding the set of data points stored.
- Location data is stored as a set of places & times, with no data relating it back to any individual user, or correlating it to other location data points from the same user.

For other Safe Paths users to be able to receive exposure notifications, without sharing their own location data, this data needs to be made accessible to other Safe Paths users. We do so in a manner that allows the Safe Paths app to detect matches on specific locations and times, while making it difficult (though not impossible) for anyone to actually view the set of locations and times that represent these “points of concern”. For details on how we do this, see below.

Locations where Diagnosed Users have been

Our 3rd constituency of concern for privacy are those businesses where diagnosed users have been.

Many businesses are already struggling as a result of the Pandemic, and there have been examples (in South Korea, for example) where cafes and shops have lost business after being associated with an infected carrier of COVID. See 4.0.3 of https://arxiv.org/pdf/2003.08567.pdf

However, while individual privacy can be protected through anonymization and pooling of location data, the same is not true for businesses.

By providing a Mobile App that can detect location-based exposure notifications, we open up the possibility for someone to monitor a business for reported COVID infection. For example, they could deliberately leave a phone running the App at the premises, and then collect this later and use the App to determine whether there had been any exposure notifications in that location.

However, while we cannot protect locations completely from this kind of monitoring, we can do our best to make it as difficult as possible. And in particular we can make it difficult to perform this kind of monitoring across large areas.

Design for Privacy-protecting Exposure Notifications

The sections above explained many of the techniques we use to protect privacy:

- Keeping uninfected users' data on their phones
- Redaction and anonymization of data before storing or publishing.

This section now explores the technical implementation that we use to make it difficult (though not impossible) to explore the “points of concern” data published by Health Authorities.

The approach that we take is approximately that outlined in section 5 of this paper as an “Intermediary Implementation” https://arxiv.org/pdf/2003.14412v2.pdf, though with some variations in the detail.

In summary:

- Data is published by Health Authorities, to allow Mobile Apps to check locally for matches with “points of concern”.
- Prior to publication, each “point of concern” is mapped from an exact latitude and longitude, to a geohash, and a 5 minute time window. This information is then hashed, using a slow consistent one-way hash algorithm, and published only in that hashed form.
- Each Health Authority publishes their data at a URL that they control, which is stored in an overall database of Health Authorities maintained by Path Check, and passed to the Mobile App when they register for Exposure Notifications from that Health Authority. This URL is not publicly advertised, or shown to users, but it can be easily determined by an attacker.
- A Mobile App can assess a given location & time for exposure by computing the same geohash, time-window and hash, and checking for a match against the published data.
- Due to the use of a slow hash, it is expensive to check a large set of locations and/or times. And because the hash is one-way, it is impossible to reverse other than by computing the hashes for every possible point in the area & time period covered.

Below we provide details of the implementation, and the reasons for these. We then assess the level of protection that this scheme manages to provide.

Geohash tile size

We use 8 character Geohashes, which resolve to rectangular tiles that are 38m x 19m at the equator, and about 27m x 19m at 45 degrees north.

The MIT paper cited above suggested a hexagonal grid, rather than a rectangular grid. A hexagonal grid offers some benefits over a rectangular grid when it comes to distance measurements, but the ready availability of geohash libraries meant that using geohashes made for a much simpler implementation.

We considered 3 resolutions of Geohash:

- 7 character = 150m x 150m
- 8 character = 38m x 19m
- 9 character = 5m x 5m.

7 character geohashes offer greater privacy protection in the event that “points of concern” data is actually decoded, as they inherently blur a location. However, we rejected 7 character geohashes on the basis that:

- They would lead to far too many false positives on exposure notifications.
- They considerably reduce the entropy of the place/time space that we are hashing, making it much less costly to decrypt the hashes by brute force methods.

9 character geohashes are appealing on the basis that:

- They greatly increase the entropy of the space, making attacks harder
- They allow for more accurate location matching.

However, our expected accuracy of location measurements is only ~10m. Allowing for this level of inaccuracy for both parties involved in an exposure notification suggests that we should be looking to match at a radius of 20m, with a resulting match area of approx 1250 sqm, or 50 of these 9 character geohashes.

With the app having to compute 50x as many hashes, we have to reduce the cost of the slow hash to compensate, and this effectively eliminates the benefits of the increased entropy.

Although we have nor ruled out a move to 9 character geohashes in future (as it would give us slightly more accurate matching, and therefore reduced false positives), we have not yet established a clear justification for the increased complexity (and it may be better to invest the effort into even more secure approaches - as described below).

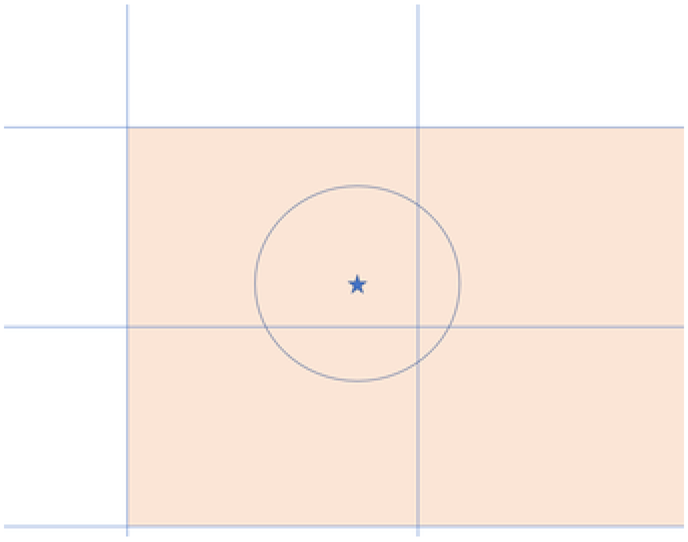
The result of the above is that our current implementation uses 8 character geohashes.

Corner Cases

Where a location is close to the corner of a geohash, and allowing for some inaccuracy in location data, there is a risk that someone else very close by, ends up reporting a different location, that maps to a different geohash.

We resolve this issue in the Mobile App by using the user’s exact location to determine a set of “nearby” geohashes, and checking them all for matches.

The definition of a “nearby” geohash is any geohash that has a point within 10m of the user’s reported location.



On average, there will be 2.3 such geohashes, though the number is any given case could vary from 1 to 6 (and possibly even more in very northern or southern latitudes).

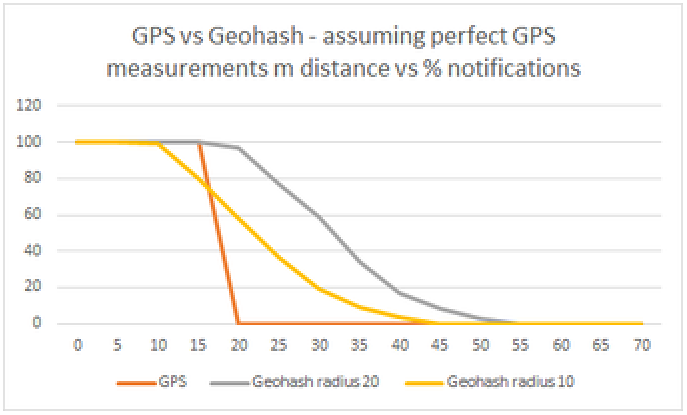
This results in a match area of around 1660 sqm at the equator, 1200 sqm at 45 degrees north, so approximately the same area as a 2om radius circle, but with a slightly less predictable shape.

The chart on the right shows how this Geohash based matching results in the probability of a match / non-match for various distances between two reported points.

The orange line shows what happens when GPS data ponts are compared directly. The yellow line shows what is achieved with a geohash tile size of 8, and a radius for “nearby” geohashes of 10m.

By moving to 9 character geohashes (as discussed above), we could bring the matching profile much closer to the orange line. This is something we may consider in future.

Time window size



Due to battery constraints (and to a lesser extent, concerns about bandwidth and storage), we record a location data point only once every 5 minutes.

There is no synchronization between the times at which two users record their location information, therefore we must consider two point to match if they fall within 5 minutes of each other.

However, with the hashing of published data, the published timestamp is itself approximated by a 5 minute time window.

To cover the case where either party might have been close to the edge of the time window, the Mobile App computes hashes for 2 different time windows, which between them cover at least +/-2.5 minutes from the actual time at which the Mobile App’s data point was recorded.

Hash Algorithm

The overall space of locations (resolved to 8 character geohashes) and timestamps (resolved to 5 minute windows) has fairly low entropy.

To provide some protection against brute force attacks over the entire space (e.g. an entire city or neighbourhood), it was important to use a slow hash algorithm.

There were two strong candidates:

- PBKDF2
- Scrypt

There was little to choose between the two, but Scrypt had the additional benefit of providing better protection against brute force attacks computed on GPUs or ASICs, due to its high memory use. We selected Scrypt on this basis.

Bcrypt was not a viable option because it uses a randomly generated salt on encryption, and we had no means of co-ordinating this random salt between two systems computing a hash.

We are in the process of tuning the cost for Scrypt, where we need to balance between:

- Providing as much protection as possible against brute force attacks
- CPU usage (and therefore battery consumption) on end-users' phones - particularly considering that for reasons of inclusivity (and because of some of the poorer countries we are targeting) we need the App to run even on low-end and old phones.
- Total time & cost to compute hashes when generating “points of concern” data sets in our Safe Places solution for Contact Tracers.

By computing hashes at the point that we record a location point on a user’s smartphone, we spread out the cost of computing the hashes, avoiding the need to compute a large number of hashes when it comes to comparing location data with points of concern.

The length of the hashes we output is 64 bits / 8 bytes.

The total entropy, across the entire land surface of the earth, and 1 year of 5 minute time windows, is about 5 x 10^16, or 58 bits. So 64 bits is plenty to make collisions unlikely over a couple of weeks, in the area covered by a single Health Authority.

Salts

For the scrypt hash, we use a single, fixed salt.

We are well aware that best practice with scrypt would be to use a long unique salt for each hash. However we have no way to co-ordinate the choice of salt between the server and the app.

An alternative design that we considered was for the Health Authority to generate its hashes with a unique salt every time they are published.

Unfortunately, this makes it impossible to pre-compute the hashes in the mobile app. In order to make it feasible for the app to compute the hashes at the point of comparison, it becomes necessary to reduce the cost of the hash function, which eliminates all the benefits of having a variable salt.

We have also considered a variable global salt (e.g one for each calendar day of data), which is not known in advance. Without a secure channel to distribute the salt to the App, the salt will become public, and therefore be equally available to an attacker. Therefore the only significant benefit will be that it will prevent pre-computation of tables. This is a modest additional protection that we are considering as a future improvement - but we may choose to focus on more substantial improvements, as described below.

Code Obfuscation

At this point in time, we have not attempted to obfuscate any of the code, or the value of the salt that we use.

We are well aware that code obfuscation can easily be reversed (or that our open source obfuscated code could simply be lifted and used to encode hashes, without even de-obfuscating it).

If we had a secure means of concealing the salty that is used, that would add considerably to the security of the published hashes, but we do not believe this is possible.

Overall level of protection

Since the space of locations and timestamps for a given Health Authority is relatively small, it has low entropy, and the overall level of protection we can provide against brute force attacks is limited.

The following table shows the costs of various hashing operations to our systems, and the costs of various attacks, on the assumption that:

- we tune the cost of our hash to take approximately 1 second, on a typical smartphone
- the servers that encode hashes for Safe Places are approximately 10x more powerful
- the servers used by an attacker are approx 100x as powerful as the smartphone.
- servers of such power could be rented in the cloud (e.g. AWS) for 10c/hour

(we are actively working to validate these assumptions)

Activity	**Smartphone CPU seconds (SCS)**	**Estimated CPU time**
Smartphone hashes 1 location	4.6	4.6 seconds
Smartphone CPU usage/day	1.3k	22 mins
Safe Places publish (100 cases)	100k SCS	3 hours
Safe Places CPU usage/day	200k SCS	6 hours
Attack on single geohash, 1 day	288 SCS	3 seconds
Attack on 1 sq km, 1 day	400k SCS	1 hour 6 mins (est. cost 12 cents)
Attack on 10km x 10km, 14 days	560M SCS	2 months (est. cost $144.00)
Attack on 3000 sq km, 14 days (e.g. a typical US County)	16.8B SCS	5 years (est. cost $4,560)

Relating this to our privacy goals articulated above

- There is no significant risk to undiagnosed users (whose data is never shared), or diagnosed users (since the data they have provided, even if it were decrypted, is anonymized and redacted).
- There is, unfortunately, a vulnerability for venues where diagnosed users have been - particularly in the case where an attacker has in mind a small area and a narrow time window, which they wish to evaluate for points of concern.

However, while such attacks are possible, it nevertheless will take significant technical skill and effort to conduct such an attack, and the rewards are fairly minimal. Our expectation is that this will lead to few such attacks will be undertaken, and that this will be a negligible channel for disclosure of information about which venues were attended by people later diagnoses with COVID.

In practise, we expect that the level of information leaked by this channel will be negligible, as compared to the level of information leaked through informal networks of local gossip.

Some Clarifying Points on Efficacy

The main purpose of this article is to present our approach to data privacy. However some of the points raised here may raised broader concerns around efficacy of location-based exposure notifications.

Without covering these issues comprehensively, here are some high-level responses to some concerns that may have arisen.

Some Notes on False Positives

Given that the time window for an exposure match extends to a total of 10 minutes, and the match area is 1600 sqm or more, including some points up to 40m away, it would be reasonable to be concerned about false positives.

While this is more related to efficacy, rather than privacy, it is worth a brief note on this topic.

Since the latest epidemiological advice on COVID is that brief contacts are not a major risk for transmission, we allow Health Authorities to set a threshold for exposure notification that only triggers when a certain number of matches with “points of concern” are detected within a set period.

Our current recommended values for these are 4 matches within 30 minutes, but this can be configured by the Health Authority, and the optimal settings here remains an area for further study.

Why not just use Bluetooth for Exposure Notifications?

It is true that Bluetooth-based Exposure Notifications promise greater accuracy than location-based exposure notifications, and avoid some of the privacy concerns that we have described on this page.

We are fully supportive of Bluetooth as a technology for exposure tracking and notification. Our Project Aurora[link] uses the Google / Apple Exposure Notifications API. We also believe there is an important role for location-based Exposure Notification systems, for the following reasons.

- Significant numbers of smartphones do not have the necessary Bluetooth capabilities.
- Location-based exposure notifications enables the particpiation for users without smartphones, or who had not downloaded a DCT App prior to being diagnosed with COVID.
- The Google / Apple Exposure Notification solution (GAEN) that uses Bluetooth does not permit the gathering of location information within the same App.

Future Work

While we believe that the solution above offers a broad level of privacy protection to the key stakeholders in the communities that we hope to serve, we are aware that it is not yet as good as it could be.

Further exploring how to protect privacy, and in particular to protect against brute force attacks over small and medium-sized areas is an area of active ongoing research at Path Check.

While plans are still under development, we believe we may be able to do better using Fully Homomorphic Encryption (FHE).

Fully Homomorphic Encryption allows computations to be performed on encrypted data, without ever knowing what the encrypted values actually are.

Using FHE, we have a design that would require every comparison of its data set against a given point of concern to require consultation with a key server, in order to determine whether there was in fact a match.

By inserting this key server onto the path, we can then use rate-limiting to block attempts to drive a brute-force comparison, while allowing legitimate users to still perform the comparisons that they need to.

Research in this field is still ongoing, but we would be happy to discuss the details with anyone who is interested, or may wish to help us in developing this additional level of privacy protection.