BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders

Jason Bay, Joel Kek, Alvin Tan, Chai Sheng Hau, Lai Yongquan, Janice Tan, Tang Anh Quy
Government Technology Agency
Singapore

ABSTRACT

TraceTogether is the first national deployment of a Bluetooth-based contact tracing system in the world. It was developed by Singapore's Government Technology Agency and the Ministry of Health to help the country better respond to epidemics.

Following its release, more than 50 governments have expressed interest in adopting or adapting TraceTogether for their countries. Responding to this interest, we are releasing an overview of **BlueTrace**, the privacy-preserving protocol that underpins TraceTogether, as well as **OpenTrace**, a reference implementation.

OpenTrace comprises the source code for an iOS app, an Android app, a cloud-based backend, and baseline signal strength calibration data. This will be made available to the open source community at github.com/opentrace-community on 9 April 2020.

1 CONTEXT

Contact tracing is an important tool for reducing the spread of infectious diseases. Its goal is to reduce a disease's effective reproductive number (R) by identifying people who have been exposed to the virus through an infected person and contacting them to provide early detection, tailored guidance, and timely treatment. By stopping virus transmission chains, contact tracing helps "flatten the curve" and reduces the peak burden of a disease on the healthcare system. Contact tracing forms an essential part of Singapore's response to the COVID-19 pandemic.

2 OVERVIEW OF BLUETRACE

BlueTrace is a protocol for logging Bluetooth encounters between participating devices to facilitate contact tracing, while protecting the users' personal data and privacy.

When two participating devices encounter each other, they exchange *non-personally identifiable messages* that contain temporary identifiers. The identifiers rotate frequently to prevent third parties from tracking users. The user's encounter history is stored locally on their user's device; none of this data can be directly accessed by the health authority.

If a user is infected or is the subject of contact tracing, they will be asked to share their encounter history with the relevant health authority with the use of a PIN. (A verification code may optionally be provided, to authenticate the health authority official's request.) Only the health authority has the ability to decrypt the shared encounter history to obtain and use personally-identifiable information to filter for close contacts and contact potentially infected users.

BlueTrace is designed to supplement manual contact tracing by addressing its key limitation: an infected person can only report contacts they are acquainted with and remember having met. BlueTrace could also allow for contact tracing to be more scalable and less resource-intensive.

BlueTrace also allows a *federated network of credentialed health authorities* to each maintain distinct user bases, while allowing for contact tracing between users from different health authority jurisdictions (see *Section 7: Federation and Interoperability*).

3 DATA PROTECTION AND PRIVACY SAFEGUARDS

We believe that even during pandemics, public health and personal privacy should not be a binary choice. BlueTrace is designed to safeguard user privacy and give users control of their data. The protocol includes the following privacy safeguards:

- **Limited collection of personally-identifiable information.** The only personally-identifiable information collected is a phone number, which is securely stored by the health authority.
- **Local storage of encounter history.** Each user's encounter history is stored exclusively on their own device. The health authority only has access to this history when an infected person chooses to share it.

- **Third-parties cannot use BlueTrace communications to track users over time.** A device's temporary identifier rotates frequently, preventing malicious actors from tracking individual users over time by sniffing for BlueTrace messages.
- **Revocable consent.** Users have control of their personal data. When they withdraw consent, all personally-identifiable data stored at the health authority is deleted. All encounter history will thus cease to be linked to the user.

4 HOW BLUETRACE WORKS

User registration and assignment of UserID

When the user of a BlueTrace-implementing app registers with their phone number, the back-end service generates a unique, randomised UserID and associates it with the user's phone number [Figure 1].

Phone numbers are the only personally-identifiable information required from the user. The phone numbers are used to contact users if they are found to have had prolonged exposure to an infected person.

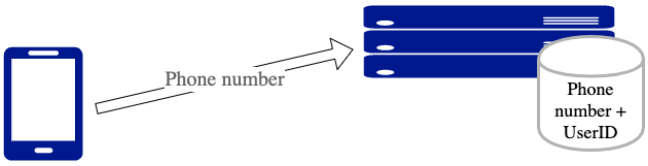

Figure 1: User registration

Alternative implementations of BlueTrace that do not require a phone number are possible. These might rely on push notification tokens to alert individual users [Section 5: Protocol Design Considerations].

Generation of TempIDs

BlueTrace devices log encounters with each other by exchanging messages over Bluetooth. To protect users' privacy, these messages cannot reveal users' identity. In addition, these messages cannot contain static identifiers, to prevent users from being tracked over time by third parties. However, when an infected user uploads these messages to the health authority, the authority must be able to obtain contact information from the messages.

BlueTrace addresses this by having users exchange temporary IDs (TempIDs). Each TempID comprises a UserID, created time, and expiry time encrypted symmetrically with `AES-256-GCM` and then `Base64` encoded [Figure 2]. Only the health authority holds the secret key to encrypt and decrypt TempIDs. Each TempID is generated with a random Initialisation Vector (IV).

The TempID also includes two encryption parameters: the IV input and an Auth Tag (for integrity checks).

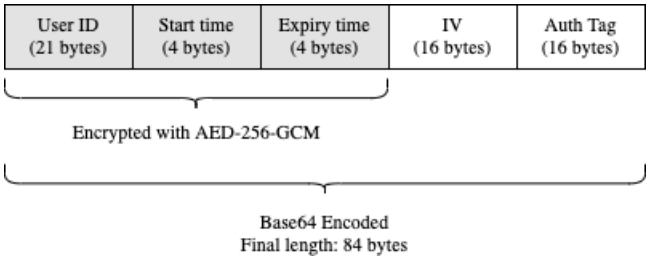

Figure 2: Format of TempID

TempIDs have a short lifetime (we recommend 15 minutes). This helps to mitigate the impact of replay attacks, by reducing the window of opportunity for exploitation. If malicious users impersonate other users by rebroadcasting their messages, they will only be able to do so for a short time before the message expires. This duration would likely be below the threshold duration of close contact, and hence not result in false positives (see *Section 8: Encounter Message replay/relay attacks*).

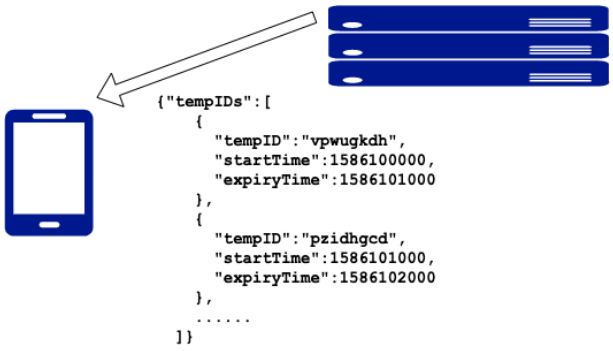

Figure 3: TempIDs sent to device

In order to ensure that devices have a supply of valid TempIDs even when the internet connection is unstable, devices pull batches of forward-dated TempIDs from the health authority's back-end service each time [Figure 3].

BLE handshake flow

BlueTrace devices exchange messages over the Bluetooth Low Energy (BLE) protocol.

In BLE parlance, devices can take on **Peripheral** or **Central** roles. **Peripherals** advertise **Services**, and **Centrals** scan for Peripherals' advertisements to connect to their **Services**. **Services** are a collection of data, such as **Characteristics**, which are specific data that can be exchanged between devices, through read and writes performed by a **Central**. The data exchanged by BlueTrace devices in each "handshake" is called an **Encounter Message**.

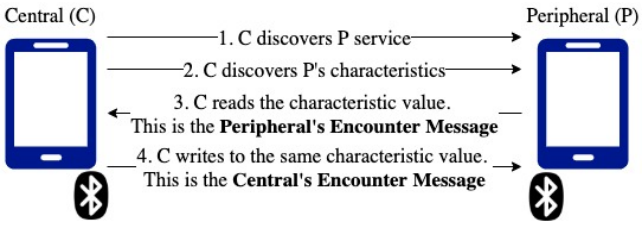

Figure 4: BLE handshake flow

Devices using BlueTrace act as both a Central and a Peripheral, and may alternate between these roles. When two devices connect, the Central reads the Peripheral's Encounter Message, and then writes back its own Encounter Message; each connection allows for a two-way exchange of data between the Central and Peripheral [Figure 4]. Allowing for two-way communications promotes symmetry and addresses the limitation where some devices (and possibly wearables) are only able to function as Peripherals.

Scanning and advertising cycles

BlueTrace devices scan and advertise on configurable cycles. Scanning occurs with a duty cycle around 15-20%, during which devices scan for other BlueTrace devices as Central. Devices may optionally introduce random jitter into the length and duty ratio of each scanning cycle to avoid lockstep behaviour.

Advertising occurs with a higher duty cycle of around 90-100%. We recommend a shorter duty cycle for scanning to conserve resources. We also recommend that the sum of both scanning and advertising duty cycles be greater than 1, to ensure that devices have the opportunity to see each other.

Blacklisting

To ensure an even distribution of Bluetooth "handshakes" with as many nearby BlueTrace devices as possible, BlueTrace devices should implement a blacklist of recently seen devices and not attempt to connect to them for the duration of the blacklist period. On both Android and iOS devices, the length of this blacklist period is between one and two scanning cycles.

Note that the blacklist can be negated by Peripherals that perform device identifier randomisation regularly. On some Android devices, this can happen extremely frequently. Such devices tend to be scanned by Centrals repeatedly, preventing an even distribution of encounters with nearby devices.

We are experimenting with different methods of preventing repetitive connections, and will incorporate recommended solutions within this document, and make the corresponding contributions to the OpenTrace reference implementation in due course.

Encounter Message

The Encounter Message is a `UTF-8` encoded `JSON`. The fields in the `JSON` differ slightly depending on the direction of communication.

The **Peripheral's Encounter Message** is advertised by the Peripheral as a Characteristic Value, so that a Central can scan for, and read it, after discovering the Peripheral and its valid Characteristic. It is in the following format (as of Version 2):

```
{
  // TempID of the Peripheral
  "id": "Fj5jfbTtDySw8JoVsCmeul0wsoIcJKRPV0
      HtEFUlNvNg6C3wyGj8R1utPbw+Iz8tqAdpbxR1
      nSvr+ILXPG==",
  // Device model of the Peripheral, to
      calibrate distance estimates
  "mp": "Samsung S8",
  // Organisation code indicating the
      country and health authority with
      which the Peripheral is enrolled
  "o": "SG_MOH",
  // Version of the BlueTrace protocol that
      the Peripheral is running
  "v": 2
}
```

The **Central's Encounter Message** is returned to the Peripheral as a Characteristic Value, that a

Central writes back to the Peripheral before closing the connection. It is in the following format (as of Version 2):

```
{
  // TempID of the Central
  "id": "Fj5jfbTtDySw8JoVsCmeul0wsoIcJKRPV0
      HtEFUlNvNg6C3wyGj8R1utPbw+Iz8tqAdpbxR1
      nSvr+ILXPG==",
  // Device model of the Central, to
      calibrate distance estimates
  "mc": "iPhone X",
  // Received Signal Strength Indicator (
      RSSI) as measured by the Central of
      the Peripheral
  "rs": -60,
  // Organisation code indicating the
      country and health authority with
      which the Central is enrolled
  "o": "SG_MOH",
  // Version of the BlueTrace protocol that
      the Central is running
  "v": 2
}
```

The main difference is that the message originating from Central contains the RSSI field. This is necessary because although the Central and Peripheral communicate in both directions, only the Central can record RSSI. Thus, the Central records the RSSI reading of the Peripheral, and then returns this information to the Peripheral so that both devices have symmetric knowledge, and so that the RSSI and device model can be used to estimate distance subsequently.

In testing, we have encountered a message size limit with some devices. This message format fits well within that constraint. If there is a need to accommodate devices with smaller message size limits, it is possible to use a byte array instead of `JSON`, and also to base64 decode the TempID.

Migrations to new message formats are possible by advertising multiple Characteristics within the Service, each corresponding to a different protocol version. This way, devices maintain backward compatibility while allowing the protocol to evolve [Figure 5].

Storage of encounter history

Both Central and Peripheral devices store each such "handshake" as an entry in its encounter history for a certain number of days (for OpenTrace, 21 days) before deletion.

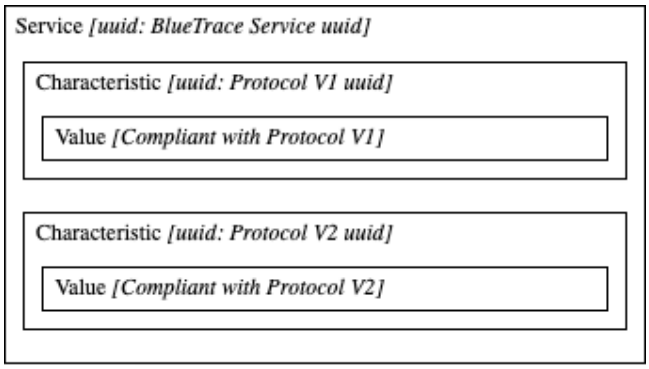

Figure 5: Protocol evolution by advertising multiple characteristics

Devices can also be configured to log when a scan is performed, to differentiate between the absence of scanning and the absence of nearby devices.

Contact tracing flow

When patients have been confirmed to be infected, health authorities ask them if they have the app installed. If they do, they are asked to upload their encounter history to the health authority [Figure 6].

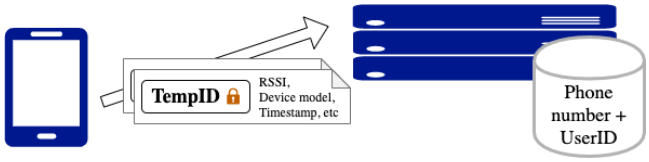

Figure 6: Upload of encounter history to health authority

To protect users and the system from fraudulent uploads, an authorisation code is provided by the health authority and entered through the app in order to obtain a valid token to transmit the logs.

Data analysis flow

The health authority decrypts the TempID for each encounter in the uploaded encounter history, in order to obtain the UserID and validity period. It then verifies that the encounter timestamp for each TempID falls within its validity period.

The health authority then filters for close contacts based on the disease's epidemiological parameters:

time of exposure (measured by the length of a continuous cluster of encounters) and distance (measured by the received signal strength reading).

In Singapore, the contact tracing process involves an interview with the patient, where the patients are asked to recall where they have been and who they have been in contact with recently. This information is used together with the BlueTrace data to adjust the proximity and duration filtering thresholds based on the patient-reported location and context.

The health authority then contacts individuals assessed to have a high likelihood of exposure to the disease, to provide medical guidance and care.

Note that this workflow can be automated and decentralised without affecting interoperability with other BlueTrace implementations. However, we do not recommend this, and have therefore not implemented it in OpenTrace. (For a further discussion, see *Section 5: Protocol Design Considerations*).

Withdrawal of consent

We believe users should be in control of their personal data and have the ability to delete this from the system. If a user withdraws consent to use their personal data, their UserID and phone number should be deleted from the back-end database. Since the phone number is the only source of identity, deleting it will render useless all of this user's TempIDs that were previously sent to other devices.

5 PROTOCOL DESIGN CONSIDERATIONS

Bluetooth vs GPS

Bluetooth and GPS contact tracing solutions were both considered. Table 1 illustrates the main differences.

Bluetooth was chosen because it is able to classify close contacts with a significantly lower false positive rate than GPS. Given that GPS accuracy decreases in indoor environments, entire shopping malls or skyscrapers would be within the margin of error of a single GPS point. Furthermore, adoption could be

	Bluetooth	**GPS**
General Approach	Devices log encounters with other devices. Infected users upload their encounter history.	Devices log their GPS location. Infected users upload their location history.
Accuracy (As a reference, widely-accepted epidemiological parameter for close contact with COVID-19 patient is 30 minutes at a distance of less than 2 metres	Able to approximate close contacts within 2 metres, by filtering encounters by signal strength. Bluetooth has a range of 10 metres in indoor environments, but RSSI follows inverse square law and drops off quickly with distance. However, calibration is necessary for maximal effectiveness as different devices transmit at different powers..	Unable to filter for proximity. Accuracy of 10 metres, which decreases in urban environments with tall buildings. Limited vertical accuracy (for floor detection) means that most people within a single skyscraper would register within the margin of error. Poor accuracy in moving or underground environments like a subway train.
Adoption challenges	Requires high adoption to be effective, because effectiveness is a quadratic function of adoption.	Requires high adoption to be effective, because effectiveness is a quadratic function of adoption unless other data sources are incorporated. Public wariness and possible alarm about tracking location data of individuals could hamper adoption.
Battery use	Low	Medium

Table 1: Comparison between Bluetooth and GPS contact tracing

hampered by the public wariness of location tracking and increased battery drain.

Generation of TempID by backend service vs on device

In the reference implementation, TempIDs are cryptographically generated by the backend service. The downside is that this requires devices to connect periodically to the internet. We account for periods without connectivity by issuing a batch of TempIDs at a time.

(An alternative would be for the UserID to be stored on the device, and for TempIDs to be generated locally using an asymmetric encryption key, with the backend service holding the corresponding decryption key. The asymmetric encryption key can be generated by the backend service and sent to the user device using registration. However, we found that this cryptographic scheme increased the computational requirement on devices beyond the OS-allocated limits – especially when in background execution mode.)

Apart from minimising on-device compute requirements, server-side TempID generation has a secondary benefit of allowing the health authority to understand adoption and usage levels of the app by logging the issuance of daily batches of TempIDs, and its potential effectiveness in epidemic control. This could then be used to inform public health policy interventions.

Centralised vs decentralised contact tracing

BlueTrace envisages a blend of **decentralised proximity data collection** and logging, with a **centralised contact tracing capability**.

Encounter messages and encounter histories are exchanged and stored in a decentralised, peer-to-peer manner, without the participation of a central server.

We defer the centralised collection and processing of data to the last possible moment—when a diagnosis of COVID-19 is made—and then provide this data to the trusted public health authority in the OpenTrace reference implementation. Depending on the prevailing trust environment within which public health institutions operate, other jurisdictions may have different considerations that may favour a similar hybrid model or one that is completely decentralised.

We see various challenges with a purely decentralised contact tracing system. Individuals falsely declaring themselves infected would cause unnecessary anxiety and panic in other users, and erode trust in the system. Some form of authorisation for users to either flag themselves as positive COVID-19 cases, or to upload encounter history, is therefore necessary to protect against abuse.

Ultimately, this will have to be provided by a credentialed health institution or healthcare worker, who may or may not be part of a public health authority's infectious disease surveillance system, but would likely have to obtain the upload authorisation code through a chain of trust rooted in a centralised public health authority. This also has the benefit of ensuring that relevant information about the epidemic and the effect and effectiveness of such contact tracing systems is provided to the public health authority, to aid in planning public health interventions.

Finally, another advantage of a centralised approach is keeping humans in the loop in making the assessment of the appropriate follow-up actions.

Human-in-the-loop vs Human-out-of-the-loop

It is possible to implement the BlueTrace protocol and have automated notification of probable close contacts of persons who have been diagnosed with COVID-19. In theory, we appreciate the privacy and scalability benefits of doing so. *In practice, our ongoing conversations with public health authority officials performing epidemic surveillance and conducting contact tracing operations compel us to recommend otherwise.*

An automated algorithm will necessarily generate both false negatives and false positives. A human contact tracer will similarly make mistakes. However, because a human contact tracer would seek to incorporate information beyond just physical proximity, he/she can correct for systematic biases introduced by a purely automated notification system.

Encounters between individuals can be classified into *close*, *casual* and *transient contacts* for epidemiological purposes, based on proximity and duration of contact. However, these classifications depend on factors such as location/environment. For example, short-duration encounters in enclosed spaces without fresh ventilation often constitute close contact, even if encounter proximity and duration do not meet algorithmic thresholds.

Since Bluetooth-based contact tracing solutions do not, by themselves, record location/environment data, this information needs to be obtained through other means – a human-led contact tracing interview.

A human-in-the-loop system is also necessary to allow judgment to be applied, given the high likelihood of pre-symptomatic transmission of the SARS-CoV-2 virus. Since time is of the essence, contact tracers may preemptively wish to trace selected second-degree close contacts of a COVID-19 patient, in cases where there is a high likelihood of exposure and infection, even if the first-degree close contact has yet to test positive. For example, there may be epidemiological value in tracing close contacts of a close relative of an infected person.

A human-out-of-the-loop system will certainly yield better results than having no system at all, but where a competent human-in-the-loop system with sufficient capacity exists, we caution against an over-reliance on technology.

Finally, the experience of Singapore's contact tracers suggest that contact tracing should remain a human-fronted process. Contact tracing involves an intensive sequence of difficult and anxiety-laden conversations, and it is the role of a contact tracer to explain how a close contact might have been exposed – while respecting patient privacy – and provide assurance and guidance on next steps.

Singapore's contact tracers are on the frontline of the fight against COVID-19; they are able to do this because they incorporate multiple sources of information, demonstrate sensitivity in their conversations with Singaporeans who have had probable exposure to SARS-CoV-2, and help to minimise unnecessary anxiety and unproductive panic. These are considerations that an automated algorithm may have difficulty explaining to worried users.

6 IMPLEMENTATION CHALLENGES

iOS background Bluetooth limitations

While the Android version of the OpenTrace reference implementation functions fully as both Central and Peripheral while the app is in both foreground and background execution modes, the iOS version of OpenTrace is bound by restrictions that iOS has on background Bluetooth functionality.

When in the background, the iOS app advertises in a proprietary advertisement format that is not part of the Bluetooth standard and thus not readable by non-iOS devices. It is also unable to scan for other BlueTrace devices in any meaningful way.

The current workaround is to encourage iOS users to keep their app in the foreground, especially when in higher-risk environments. Within the OpenTrace reference implementation, we have implemented a "power saver mode", where users can flip the phone upside down to dim the screen so the app uses less battery power while in the foreground. Users, particularly inactive users, also receive push notifications to remind them to use the app, especially during commuting peak hours. The app also prompts the user if inadequate permissions are granted or Bluetooth is turned off, resulting in the app being unable to function normally.

Difference in transmission power across devices

BlueTrace uses RSSI readings to approximate distance. However, through tests of devices in anechoic chambers, we have established that the variance in transmission power across popular mobile devices can be as large as 30 dB (1000x). During testing, we have also discovered that transmission power varies little between different devices of the same model and is minimally affected by mobile phone cases.

In order to account for this difference, we have taken reference signal strength readings for popular mobile devices in Singapore. We use this to calibrate RSSI readings when classifying encounters by proximity.

We have shared this data at github.com/opentrace-community. We invite developers and handset manufacturers to contribute to this, so that it can serve as a universal calibration table of device transmission powers for any Bluetooth contact tracing solution.

7 FEDERATION AND INTEROPERABILITY

Federation is a common and natural extension of national systems and BlueTrace welcomes collaboration with the international community to facilitate community-driven cross-border contact tracing. BlueTrace was designed with interoperability in mind while maintaining flexibility for adopters of its protocol. Where possible, the protocol allows health authorities to customise and adapt the protocol to suit their use cases.

Guiding Principles

BlueTrace's guiding principles on federation and interoperability:

- Each health authority should be allowed to administer their own set of users separate from

other authorities. The user identity and contact information belonging to users of one health authority should never be exposed to another health authority.
- Each health authority can use its own algorithm for generating TempIDs and determining the validity period of the TempID. The TempID should allow the health authority to obtain the associated user's contact details.
- Each health authority is responsible for storage and protection of the users' identifiers and encounter history shared.
- Each health authority's mobile client app must perform communication exchanges using the BlueTrace Encounter Message format.
- OpenTrace has a set of default configurations for scanning and advertising cycles, but each health authority has the flexibility to configure the scanning and advertising cycle as it deems fit.

Registry of BlueTrace Health Authorities

A registry of BlueTrace Health Authorities consolidates the list of international participating authorities. The registry contains information about the participating authority such as name, organisation code, contact person details and an endpoint to allow anonymised information to be exchanged between authorities. The organisation code which is sent as part of the encounter exchange message follows the format: ISO-3166 country code (2 characters) followed by an organisation unit (3 characters) with an underscore separator, e.g. SG_MOH. In countries where there are multiple health authorities, the organisation unit can be used for intra-country federation. Interested health authorities will need to write to BlueTrace at info@bluetrace.io, before being added to the global registry. BlueTrace recommends that only a single Health Authority BlueTrace app be installed and activated on a user device for maximum effectiveness.

All Health Authorities that are part of the BlueTrace registry are required to implement the following interfaces:

- Exchange TempID for PseudoID
- Be notified of PseudoIDs that have close contact

Generation of TempIDs

BlueTrace maintains interoperability while preserving flexibility for each health authority [Figure 7]. Each authority has its own user base, datastore, and algorithm to generate TempIDs for its users. BlueTrace does not limit the information collected during registration as long as the user can be traced back to a valid phone number or can otherwise be alerted. This could be through a push notification. The two different mobile clients communicate via the BlueTrace Protocol and transfer Encounter Messages. Each Encounter Message received by the device is then logged and stored.

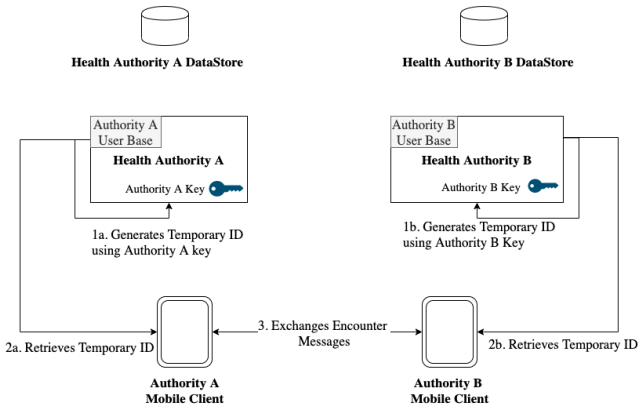

Figure 7: Interoperability between two health authorities

Processing of BlueTrace encounter history across health authorities

When a patient is diagnosed with COVID-19, the patient will be approached by the health authority to upload his data. The data which contains TempIDs and records belonging to other authorities as well as its own is then processed by the backend. The differentiation is done through the organisation code indicated in the Exchange Message.

The health authority refers to the registry of BlueTrace health authorities and forwards the TempID and timestamp to the endpoint corresponding to the organisation code. The TempID will be validated by ensuring that its timestamp falls within its validity period. The endpoint then returns a PseudoID. The PseudoID allows correlating to a unique individual for analysis in place of a TempID which changes frequently. It could be a hash of the user's UserID or a randomly generated unique identifier that is mapped to the user's UserID. Once the PseudoID is assessed to be a close contact of the infected patient, the foreign health authority, which issued the PseudoID will

be informed of the close contact period and duration, and can then follow-up as necessary. The process is illustrated in Figure 8 below.

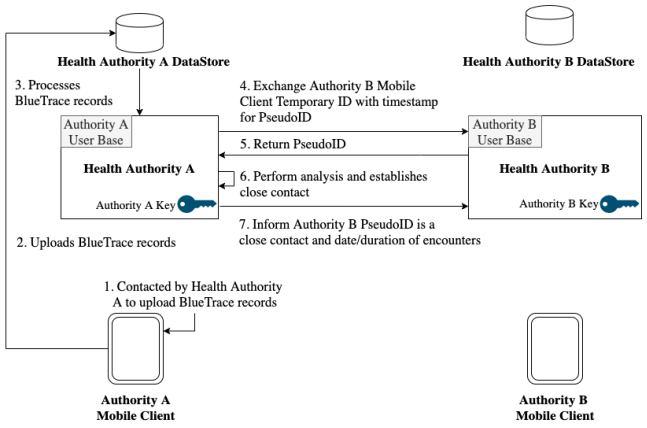

Figure 8: Upload and processing of BlueTrace records

8 SECURITY CONSIDERATIONS

Encounter Message replay/relay attack

BlueTrace protocol relies heavily on the exchange of messages through Bluetooth. This makes it susceptible to replay and relay attacks as an attacker has free access to capture the message being transmitted from a BlueTrace user's mobile device. The attacker can replicate the message (but is unable to modify the TempID) and replay/relay it across multiple locations to make it appear as if the compromised user had close contact with many other devices. BlueTrace minimises this attack vector for replay (but not relay) attacks by reducing the validity of each TempIDs to 15 minutes (which strikes a balance between threat mitigation and computation intensity). If an expired TempID is collected by a BlueTrace user, when it gets uploaded to the backend, the backend service will reject the record after checking the timestamp and validity of the TempID. In addition, the attacker will need to stay within BLE range continuously, in order to capture the latest Encounter Message from the BlueTrace user.

Ultimately, protecting against a replay/relay attack is performed not through a technical solution, but through a process solution. In the Singapore implementation, a human contact tracer will corroborate the circumstances under which an encounter has occurred, when contacting the flagged close contact, as discussed in "Human-in-the-loop vs Human-out-of-the-loop", above.

Bluetooth vulnerabilities

Many smartphone users today use Bluetooth to connect their phones with peripherals such as smart watches, headphones, etc. While it is unlikely that the use of a BlueTrace app by itself introduces additional vulnerabilities, vulnerabilities are occasionally discovered in the underlying technology that BlueTrace depends on, i.e. Bluetooth. These vulnerabilities have to be patched at the operating system-level, and we therefore urge users to ensure that their operating systems are regularly patched. BlueTrace apps may consider notifying users if an outdated operating system is detected, in order to prompt users to update them.

9 LEGAL CONSIDERATIONS

We note that data protection and privacy regulations differ from country-to-country. Health authorities that wish to deploy a BlueTrace-implementing app, whether built on top of OpenTrace or not, should seek separate legal advice on the appropriate consent mechanisms and data protection provisions in the design of the specific implementation of BlueTrace that is contemplated. Nothing in this white paper or protocol specification should be construed as legal advice in any domestic or international context.

10 CONCLUSION

We hope that our description of the BlueTrace protocol, with occasional references to how it is being implemented in Singapore, provides insight to others seeking to deploy Bluetooth-based contact tracing solutions in their own communities. We have documented the protocol and system design choices with a view to enabling globally inter-operable community-driven contact tracing. These will necessarily have to be adapted to the prevailing domestic context for each BlueTrace-implementing system. Bluetooth-based contact tracing is not a silver bullet for dealing with the COVID-19 pandemic; it must ultimately co-exist and support the pandemic response plans and processes of the public health authorities guiding us through these difficult times.

We welcome suggestions on this white paper and protocol at info@bluetrace.io.