STRIKING THE BALANCE: EFFECTIVE YET PRIVACY FRIENDLY CONTACT TRACING

Giuseppe Garofalo
imec-DistriNet, KU Leuven
giuseppe.garofalo@kuleuven.be

Tim Van hamme
imec-DistriNet, KU Leuven
tim.vanhamme@kuleuven.be

Davy Preuveneers
imec-DistriNet, KU Leuven
davy.preuveneers@kuleuven.be

Wouter Joosen
imec-DistriNet, KU Leuven
wouter.joosen@kuleuven.be

Aysajan Abidin
imec-COSIC, KU Leuven
aysajan@kuleuven.be

Mustafa A. Mustafa
University of Manchester & imec-COSIC, KU Leuven
mustafa.mustafa@manchester.ac.uk

ABSTRACT

Successful contact tracing effectively facilitates the fight against pandemics of highly contagious diseases such as COVID-19. Existing efforts either rely on effective yet privacy-invasive surveillance infrastructure, or focus on privacy-preserving decentralised solutions which may limit their effectiveness. The former collects vast amounts of sensitive data such as identity, location and social interactions of every user, which allows function creep. The latter relies on users' willingness to share their risk scores with authorities, which limits their ability to quickly identify people at-risk and to run analytics. We propose a practical solution that aims to strike a balance between functionality and privacy: one that does not collect sensitive information, such as, location data, while at the same time allowing effective tracing and notifying the close contacts of infected users. To protect users' privacy, our solution uses local proximity tracing based on broadcasting and recording constantly changing anonymous public keys via short-range communication, for example, Bluetooth. These public keys are used to establish a shared secret key between two people in close contact. These three keys are then used to generate two unique per-user-per-contact hashes: one for infection registration and one for health status query. These hashes are never revealed to the public. To support functionality, risk score computation is performed centrally, which provides the health authorities with minimal, yet insightful and actionable data. Data minimization is achieved by the use of per-user-per-contact hashes and by enforcing role separation. In our design, the health authorities and the GPs act as proxies, while the matching between hashes is outsourced to a third-party, i.e. the matching service. This separation ensures that out-of-scope information, such as social interaction within the population, is hidden from the health authorities and, at the same time, the matching service does not learn sensitive information about the users. Our solution requires a degree of trust in the entities involved that is considerably lower w.r.t. centralised alternatives.

1 Introduction

The COVID-19 pandemic is currently (early 2020) holding the world hostage. As the SARS-CoV-2 virus is mainly transmitted among people via close contact with infected persons and as there is no vaccine developed yet, the best countermeasure to stop the spread of the disease is believed to be social distancing, which is mainly enforced through lockdowns. Lockdowns assume that everyone is potentially infected and encourages people to stay indoors as much as possible, effectively limiting social interaction to people's own household bubbles. This countermeasure, however, has an enormous cost, as normal social life is halted, the psychological and economical damages are immeasurable.

Due to COVID-19's long incubation time, i.e. between 3 and 13 days [1], infected people can spread the disease unknowingly. This is in sharp contrast with the other recent highly infectious disease Ebola, where people were bedridden shortly after infection. Thus, to reboot normal life while keeping the virus in check, there is a need to warn (potential) asymptomatic people (i.e. infected people with no obvious symptoms) early so that they can take appropriate measures, such as self-quarantine and testing. To this end, contact tracing has proven to be highly effective [2]. In contact tracing, the chance of infection is calculated based on recent encounters. When the chance of infection exceeds a certain threshold, the person in question is warned and urged to take appropriate measures.

Tracing populations' contacts, however, has enormous privacy implications. Some regimes implemented contact tracing on top of their already in place surveillance systems, thereby violating the privacy of all of their citizens or at least of the infected population. In Europe, the importance of privacy is better understood, but even here, the fear of the common enemy seems to empower the political elite to deploy such systems rapidly without fully addressing users' concerns for privacy. It is worth noting that the European Data Protection Board (EDPB) stands for a solution that ensures respect for users' fundamental rights [3]. This is independent of the trade-off between centralised and decentralised solutions, as long as the processing and storage of data are in line with the purpose of the contact tracing solutions.

Fortunately, there are already existing efforts to design privacy-preserving contact tracing systems [4, 5, 6, 7]. Most of these systems rely on a local protocol to broadcast frequently changing pseudonyms. Upon encountering a pseudonym, it is registered locally. When someone is tested positive for the disease, the test lab or the GP (general practitioner) sends the pseudonyms of the patient (or the seed of the pseudonyms) to a central server, which in turn broadcasts them to all users. Each user then locally calculates their risk score by comparing the received pseudonyms of the infected users with the pseudonyms of their close contacts stored on their phone. Although these systems protect users' privacy in many aspects, they still have some limitations: (1) the privacy of infected users is not guaranteed towards other users as they can (with some effort) link these pseudonyms to real people and (2) by default do not support health authorities with the necessary data to effectively study the disease, evaluate the impact of preventive measures and decide upon resource allocation.

In this paper, we address the aforementioned limitations by proposing a privacy-centric architecture that allows to combat the virus more effectively than the existing fully decentralised systems, while still protecting users' privacy. Contrary to existing efforts, our solution outsources the risk calculation in a secure and privacy-friendly way such that:

- Close contacts are registered with unique per-user-per-contact hashes that are hidden from the public as this list is sensitive information.
- The information used during infection registration and health status query are decoupled: each contact is registered with two different hashes, one used by the infected users for infection registration, while the other one is used by the other users for querying their health status.
- Additional functionality can be achieved: more precise risk score calculation; analytic capabilities to study transmission dynamics and to map the spread of the disease; and the ability for a central health authority to learn about potentially dangerous new infections.

The rest of the paper is organised as follows: Section 2 provides a comprehensive study of the already in place contact tracing systems as well as candidate systems for deployment. Section 3 specifies the key functional, security and privacy requirements for contact tracing solutions. Section 4 proposes our privacy-centric architecture that allows to combat epidemics effectively. Section 5 analyses our solution in terms of security and privacy. Section 6 discusses the properties of our solution w.r.t. existing decentralised and centralised solutions. Section 7 concludes the paper.

2 Existing efforts

An overwhelmingly long list of solutions have been proposed in an effort to curb the spread of COVID-19. We sum up the efforts of governments, private companies, and public institutes, focusing on best practices and privacy threats.

2.1 Solutions based on surveillance systems

Solutions relying on a central authority might be highly privacy-invasive for the end-user. Disclosing the identity of the user allows for extensive tracking and, eventually, leads to mass surveillance. Such solutions, however, empower central authorities to produce a fast response and limit waste of resources. Below we review the systems developed by countries that paved the way to digital contact tracing in the early phases of the epidemic.

China. China (PRC) has been the first country to adopt technology to track infections and curb the spread of COVID-19. Amongst many apps, *Health Code* by Alipay is the most widely adopted. It shows a QR colour code that describes the

risk of having contracted the virus: green is for low-risk individuals, while yellow and red enforce a period of isolation of 7 and 14 days, respectively [8]. By installing QR code readers at gathering spots, PRC enforced strict access control policies, e.g. only green-labelled users are allowed to visit supermarkets and access transport hubs. This system is highly effective but the lack of transparency raises questions. A recent report discovered how the app establishes a direct connection between the client and the police department at check time, by-passing the government central node and sharing GPS location and a unique identifier [8]. Additionally, the app allows to perform manual interviews by scanning the ubiquitous QR codes. By reporting on daily movements, people integrate GPS data with manual recordings, an instrumental measure to identify novel outbreaks [9].

Yet, it is not clear how the algorithm assigns the colour code behind the curtains. It is plausible that location data alone did not drive the success of the operation. PRC is considered to run a highly-sophisticated surveillance system in the north-west part of the country that is being used to repress ethnic minorities such as the Uighurs, a Turkic ethnic group [9, 10, 11]. This framework integrates several sources of fine-grained data, including CCTV recordings equipped with facial recognition AIs, biometric measurements, and travel history [9]. Ultimately, private companies can be asked to share sensitive data about their customers, e.g. credit card transactions, to boost up the system accuracy [12].

The strict travel restrictions enforced by the government, in conjunction with the rapid spread of virus, has bolstered anxiety among citizens. To avoid creating panic, a *close contact detector* has been developed by PRC [13]. It allows for a user to check the current status of three people, by using their unique IDs, to discover potentially infectious encounters. The implementation of the app as a plug-in of popular social network WeChat, and payment system Alipay, made it widespread among the population, therefore increasing privacy concerns. Despite being considered as a successful example of digital contact tracing platform [2], this solution would find an insurmountable hurdle in the European privacy law framework.

South Korea. Similarly to China, South Korea's contact tracing solution allows to track only infected people [14]. By exposing data through a web interface, they enabled the proliferation of third-party apps that show the whereabouts of confirmed cases in a given time window [15]. Users can then autonomously check whether they encountered infected people and report themselves as potential carriers. Privacy implications can be devastating for the public [16]. Despite being anonymised, a precise location history log leads to the easy re-identification of the patients, also revealing sensitive information. GPS location was integrated by a more extensive digital footprint, including credit card transactions and CCTV recordings, to objectively verify users' claims [17]. In this way, the government can easily identify new infections and their close contacts.

Their solution is undoubtedly highly invasive for (potentially) infected people. However, they relied on the honest, scientific reporting to build a motivated population [14]. South Korea adopted an extensive testing policy since the early developments of the disease, which is considered one of the most important drivers for the success of digital contact tracing [2]. Quarantined people are also being tracked by the government which enforces the installation of a dedicated app that, periodically, shares the users' absolute location [9]. As an additional mean of communication, SMSs are used to notify people about high-risk venues and novel outbreaks [9].

Singapore. The government of Singapore integrated manual interviews and surveillance recordings with a contact tracing app, Trace-Together [18]. Unlike previous solutions, the underlying protocol, i.e. BlueTrace [19], relies on proximity instead of absolute location: by using Bluetooth signal strength as a distance metric, they capture the relative location of a pair of users. The use of Bluetooth Low Energy (BLE) reduces the effect on power consumption, allowing for a continuous advertisement to nearby devices. They define a close contact in terms of distance ($\approx$ 2 meters) and timespan ($\approx$ 30 minutes). During the advertisement phase, a user broadcasts a packet containing the following information: a randomised identifier, a timestamp, and device-related information. This package is collected with its signal strength, and it is encrypted and stored locally for up to 21 days. For diagnosed users, the list of packets originated by close contacts is decrypted and shared with the government. Unfortunately, randomising the identifier is not enough to protect against linkage attacks. By linking the users to their pseudonyms, the government can build a social interaction graph, eventually de-anonymising infected people and their close contacts. Increasing the frequency at which identifiers are refreshed is essential to protect the end-users against eavesdroppers, who could re-identify infected users among their close contacts by means of a timestamp [20]. It is worth to mention that their system provides additional privacy guarantees for the non-infected users than the solutions deployed by China and South Korea. Using Trace-Together, users who have not come in close contact with high-risk individuals do not disclose any information with the government. However, they know the identity of the positive cases and their close contacts.

Contact tracing did not avoid a partial lockdown of the country, as the Singaporean Prime Minister pointed out in an official statement: "[...] despite our good contact tracing, for nearly half of these cases, we do not know where or from whom the person caught the virus" [21]. Despite a general sense of trust in the government practices, only one out of four Singaporean have downloaded the app (as of May 10, 2020), which is considered insufficient for digital contact

tracing to become effective. The remaining privacy issues and OS compatibility issues leading to rapid battery drain were considered to be the main factors that hindered the adoption of the app [22].

Additional means of tracking are being used for quarantined people, which are checked via random calls (twice a day) and SMSs. They can be asked to issue a proof of isolation by taking a picture at home or following a link [9]. This represents once more a major (and avoidable) privacy invasion.

2.2 Proposed privacy-preserving solutions

A number of approaches have emerged that steer clear from a centralised solution. They aim to empower citizens as responsible contributors, instead of treating them like suspects. As the boundary between privacy and functionality is often blurred, each of them comes with its flaws that are worth an in-depth analysis.

Boston University (BU) [4]. Each participating user's mobile phone constantly broadcasts using its short-range communication interface (e.g., Bluetooth) a random number (token) that changes every few minutes; simultaneously, each phone records the random tokens received from neighboring phones. As soon as a user is proven to be infected or potentially contagious during a certain time period, a medical personnel requests the tokens that the user's phone transmitted during that time period and uploads them to a public registry. Other users' phones then at regular intervals download the tokens from the public registry and check if they match the tokens they have collected. If a match is found, the phone owner knows that they has been in close contact with an infected person, hence they can follow the official guidelines. The scheme only notifies the user of the existence of close contact, not of the time or location of the contact. The public registry only holds unnamed random tokens.

DP-3T [5]. It builds on the above-mentioned basic approach by introducing the following features. 1) A participating user's phone also records course time window (morning/afternoon/evening and date) along with each random tokens (i.e., pseudonymous identities) received by the phones of users of close contact. 2) Once a user is proven to be infected, with the assistance of a medical personnel, the user's phone sends the seed used to generate the random tokens to the backend. 3) To combat traffic analysis, the users' phones periodically upload data at predefined times - the real seed in case of a user proven infected and dummy messages otherwise. The backend stores the seeds of the infected users. Again, several times a day, the backend broadcasts the newly registered seeds of infected users. Other users' phones, once received the seeds from the backend, generate the random tokens from the seeds and compare if they have any of these random tokens stored on their phones. If matches are found, based on the parameters (course time window) of these matches, the user's phone calculates a risk score. If this risk is above a certain threshold, the user is notified and a potential course of action is displayed on their phone. Moreover, due to the recorded course time window, the user also knows the date as well as the approximate time of the close contact (e.g. morning).

Covid Watch (w/ Stanford) [6]. Akin to (1) the discrete number of Bluetooth contacts and signal strength are instrumental to recognise close contacts and estimate their duration. Each device keeps a log of random identifiers, both received and transmitted. On top of the previous solutions, they automate the upload of the list of random numbers: if a person is proven to be infected, they receive a permission number by heath authorities to upload their log. The server broadcasts the pseudonyms of infected users to be matched locally by each user. At the time of writing, the initial plan of integrating anonymised GPS location to derive useful statistics, e.g. HeatMaps, has been dropped.

Private Kit: Safe Paths (MIT) [7]. The app downloads the encrypted trail of the infected patients derived from three sources: Google location history, GPS recorded by the application itself, and manual interviews. This solution guarantees that non-infected people cannot be eavesdropped by nearby devices, since data stay local. However, GPS location discloses much more information than proximity traces. To prevent leakages, they allow the health authority to blur and redact trajectories upon disclosure, an operation that can destroy the utility of data while allowing for re-identification of the positive cases and their close contacts (individuals at risk). This solution does not involve a third party other than the health authorities, therefore reducing risks given by centralised solutions.

ROBERT [23]. This is one of the proposals by the Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) consortium. In ROBERT, users are first registered into the system at the backend server with their persistent unique identifiers (PUID) of their devices. Then, the backend generates and distributes a list of ephemeral IDs for each user. The users' devices broadcast (e.g. via BLE beacon) their ephemeral IDs so that nearby users' devices locally store the observed ephemeral IDs. When a user tests positive, the user uploads their contacts' ephemeral IDs to the backend and their account is deactivated. Additional meta data – such as time of contact, BLE signal strength and transceiver power – is uploaded by the infected user depending on what is required by the algorithm for risk score calculation. A user performs an exposure request by sending their own ephemeral IDs along with their message authentication codes and timestamps for integrity checks. The backend calculates the risk score for the requesting user and send them a binary response: if the risk score is above a predefined threshold, the user is flagged notified and asked to follow a protocol.

In summary, countries that implemented contact tracing have limited the rate of infection while minimising disruption for the broad society. Quick quarantine enforcement of individuals at risk and gathering of essential information has been instrumental to their approaches. However, the lack of transparency about their closed system led to criticisms and dismay amongst privacy advocates. More open protocols have been proposed by researchers. The proposed solutions can be separated into two categories: centralised and decentralised. The former generate ephemeral IDs and performs risk score calculations centrally. Whereas the latter do both operations locally on the user device. Some centralised approaches – such as ROBERT – may not achieve the needed anonymisation guarantees, thus becoming highly vulnerable to linkage attacks. Decentralised solutions – such as DP-3T – target privacy as their basic goal. They weaken the central health authority that plays a coordinating role in combating an epidemic and rely on the users to behave responsibly. While this approach is better suited to the European regulatory framework, it might be insufficient to tackle challenges originating from a world sanitary emergency.

3 Striking the balance between functionality and privacy

In this section, we attempt to determine a good balance between functionality and privacy which contact tracing solutions should aim for. First, we define the high-level goals of a contact tracing solution, before translating these goals to concrete functional, security, and privacy requirements.

3.1 Goals

The European Commission (EC) states that the goals of contact tracing are to **avoid transmissions** and **learn transmission dynamics** [24]. The latter goal can be further extended to a third goal - **map the spread of the disease**. Below we elaborate more on these three goals.

- **Avoid transmissions.** The primary goal of contact tracing is the eradication of infectious diseases by avoiding transmissions among people. More specifically, contact tracing aims to identify at-risk individuals based on their recent physical contacts, i.e. people who have a high chance of having contracted the disease due to an encounter with an infected person. Identification of at-risk individuals allows to take the appropriate preventive and reactive measures, i.e. self-quarantine and directed testing.
- **Build transmission dynamics.** The secondary goal of contact tracing is obtaining new insights on how the disease is transferred. By aggregating data on the occurrence of infections, patterns can be found that aid authorities to decide upon preventive measures at specific locations or nationwide.
- **Map the spread of the disease.** The analytics capabilities used for the secondary goal can be extended to gain insights on the spread of the disease. Currently, governments are already actively developing tools that allow to track the magnitude of the epidemic, and determine its geographic spread. The first allows them to evaluate the impact of current measures, while the latter permits a better distribution of resources. Contact tracing can easily facilitate these needs.

3.2 Functional requirements

The EC acknowledges [24] that to achieve the aforementioned goals contact tracing solutions should abide the following functional requirements:

(F1) The system should support a **realistic contact model** that can be updated if the understanding of virus transmission changes. As the European Center for Disease Control (ECDC) currently defines the riskiness of close contact based on distance (≈ 2 meters), duration (≈ 15 minutes), and historical importance (incubation period on average five to six days), this requirement translates to:

  (a) Duration of contact should be measured at least in 15 minute chunks.

  (b) Distance accuracy should be around 1 meter.

  (c) Contacts should be stored for 14 days.

(F2) Each individual's device should maintain a **complete list of contacts**, preferably in an irrefutable way.

(F3) At-risk individuals should be **notified**.

(F4) **Health authorities** should **learn** the **risk scores** of users.

(F5) The system should be **scalable**, work **across borders**, and be built using **existing technology stacks**.

3.3 Security requirements

The functionality of contact tracing is threatened if fake contact information is provided on a large scale. This would endanger all three goals. More specifically, to ensure that the contact tracing solution functions correctly, it should satisfy the following security requirements:

(S1) **Infection integrity.** The system should be defended against people falsely claiming they are infected.

(S2) **Contact integrity.** Contact events that impact users' risk score should correspond to real physical encounters. Similarly, insights w.r.t. transmission dynamics or the spread of the disease cannot be based on false contacts.

(S3) **Notification integrity.** A user should receive a notification that they is at risk if and only if they had close encounters with infected users.

3.4 Privacy requirements

Europe adheres to data minimisation and privacy-by-design principles and strives to be GDPR compliant. In the contact tracing context, the three most important threats identified are false positives, behavioural profiling of infected users by GPs and heath authorities, and de-anonymisation of users [25]. Therefore, contact tracing solutions should respect individuals' privacy by collecting and processing as little private data attributes as possible. From the guidelines of the EC [24] the following private user attributes can be identified:

- **Personally Identifiable Information (PII).** This is any permanent or semi-permanent identifiers such as names, phone numbers, addresses, e-mail addresses, social security numbers, credit card numbers, etc.
- **Health status.** This status consists of the ground truth of whether an individual is infected or not as well as their risk score of being infected as a result of the individual's close contacts with infected people.
- **Social interactions.** These constitute of information about each individual's encounters with other individuals as well as the timings of these encounters, i.e. population's social interaction graphs.
- **Personal trajectories.** These are the trajectories individuals follow, i.e. any form of location data, which could also be coupled with timing data.

If all of the above listed personal attributes are obtained by any authority on a large scale, the authority would obtain big brother-like powers. Alternatively, obtaining this information on a small local scale can lead to adverse consequences for specific individuals who can easily be de-anonymised.

In general, there are three type of entities who might be interested in accessing the aforementioned private attributes of an individual: other users, health authorities and external entities observing the system. Other users and external entities should not learn any of these private attributes. Health authorities, however, would benefit of knowing the health status of their population (i.e. infected, at-risk, not-infected) to effectively achieve the goals of contact tracing. Additionally, health authorities could benefit from knowledge on the user's PII (e.g. a phone number) so that they can personally notify the user and perform targeted testing. This would be more reliable than a notification generated locally on the user's devices as well as it would speed-up resource allocation for testing. We summarise the above in Table 1. Hence, to protect users' privacy, contact tracing solutions should satisfy the following privacy requirements:

(P1) **Health status confidentiality.** Only the health authorities (including the GPs) should be able to link the PII and the health status of individuals.

(P2) **Infected user privacy.** At-risk users should not be able to identify which of their contacts affected their risk score.

(P3) **User location privacy.** No one should be able to track individuals, i.e. to obtain their location/trajectories.

(P4) **Interactions privacy.** No one should be able to build social graphs of individuals' contacts. These include *global graphs* (i.e. interactions of all users of the system) and *local proximity graphs* (i.e. interactions of subsets of users – for example infected users).

(P5) **Voluntary participation.** Participation to contact tracing should be voluntary.

4 Proposed solution

This section describes the architecture of the solution we put forth. Our solution comprises four phases: (1) bootstrapping, (2) local proximity sensing, (3) infection registration, and (4) health status query. In the following we first give an overview of the solution, thereafter the details about each phase are presented.

Table 1: Overview of who should know what private attribute of an individual.

Private attribute / Entity	Other Users	Health Authorities	External Entities
Health status	✗	✓	✗
Personally identifiable information	✗	✓/✗	✗
Trajectory	✗	✗	✗
Social interactions	✗	✗	✗

4.0 Overview

In our scheme, there are four main actors that interact in four phases. The four main actors are:

- *Users* are voluntary participants who get notified of being at risk based on their close contacts.
- *General Practitioners (GPs)* are verified medical professionals who perform a medical diagnosis based on a medical test.
- *Central Health Authority (CHA)* is the entity coordinating the protocol. It also aims to gain new insights in transmission dynamics and to accurately map the spread of the disease.
- *Matching Service (MS)* is an external party in charge of performing matching on behalf of the CHA, in order to enforce separation of concerns.

Below we briefly describe the four phases.

1. **Bootstrapping.** The necessary communication channels are set up by downloading the contact tracing application that contains all the required (crypto) parameters. GPs set up a privileged account that allows them to register infections. Users create an account with the CHA, and optionally provide some PII. From then on, they participate in contact tracing and get notified if they are at risk of having contracted the disease due to their encounters.
2. **Local proximity sensing.** Users broadcast frequently changing anonymous public keys. Upon encounters with other registered users, they calculate two identifiers for that specific contact: one for infection registration, and one for querying. We name these identifiers, report hash and query hash, respectively. We use two separate identifiers per contact to better defend private interactions, as now no two users will report the same contact identifier during registration of infections. The same holds during health status query, i.e. no two users will query with the same contact identifier.
3. **Infection registration.** If a user suspects infection, they visit a GP. The GP conducts a medical test. If the user is diagnosed with the disease, the GP uploads (via the CHA) the user's encrypted report hashes to be included in the list of hashes of infected users. This list of hashes is build by the MS. The CHA and GP fill the role of an aggregator such that the MS never learns which hashes belong to the same user. Furthermore, the CHA and GP never learn the hashes.
4. **Health status query.** A user checks their health status by querying the CHA with their query hashes and some metadata about the contacts. The CHA forwards this request to the MS, who checks for matches with the list of hashes of infected users. The CHA is informed on what query hashes matched. The CHA then calculates the risk score for the user based on the metadata it received. The user is notified about their health status. Special care is taken such that the CHA never learns the exact hashes and the MS never learns what hashes originate from the same users.

4.1 Bootstrapping

In the bootstrapping phase the necessary (crypto) parameters are exchanged, and medical personnel and users are registered. Fig. 1 illustrates these three actions.

Our system relies on a Diffie-Helman (DH) protocol to locally exchange shared secrets between users upon close physical encounters. The process allows to compute unique identifiers for every contact, the exact protocol is described in Section 4.2. To calculate these contact identifiers some public parameters need to be agreed upon. The parameters are chosen by the CHA and are put in the application by the developer in the form of a certificate kit. The certificate kit can be verified as it is signed by a root of trust. Similarly, a certificate kit with the public key of the MS is embedded in the app. The MS relies on asymmetric key cryptography to obtain contact identifiers during both infection registration

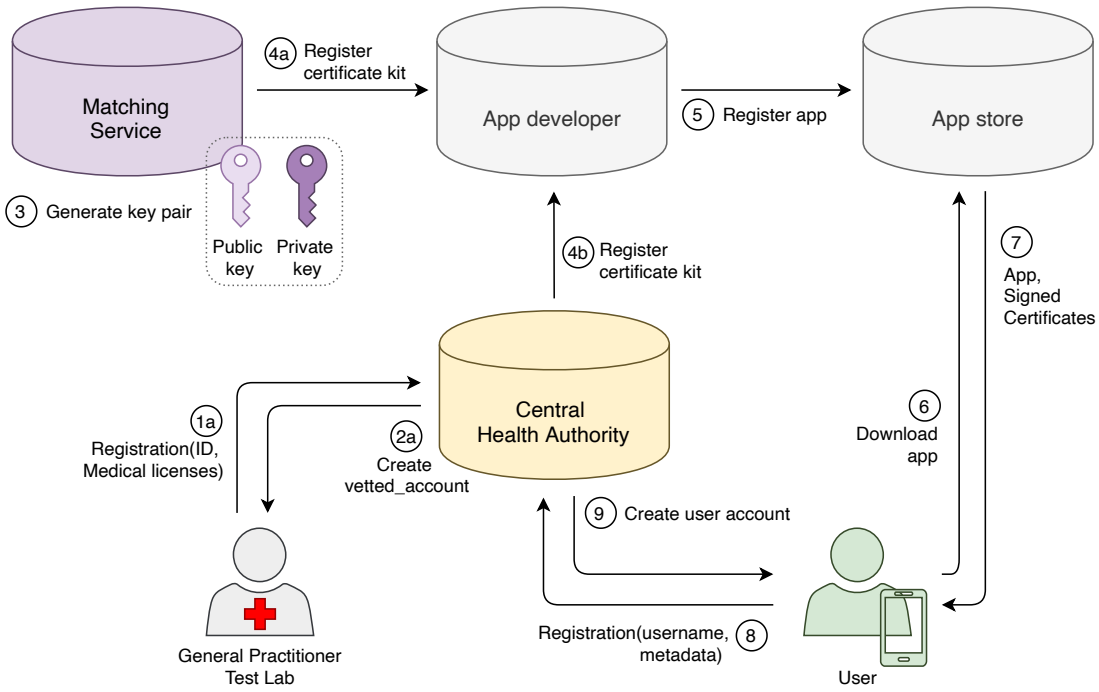

Figure 1: System bootstrap phase – necessary (crypto) parameters exchange, medical personnel and users registration.

and health status querying, without revealing them to the GP, CHA or any other external parties. The exact process is explained in Section 4.3 and Section 4.4.

To satisfy *(S1) Infection integrity*, registration of infections can only be done by authorised medical personnel, thus, a privileged account is needed. Medical experts enroll for such an account by registering themselves with the CHA. This registration process requires proof of the necessary medical expertise. This process could piggyback on already in place health systems where most of the authorised medical personnel is already registered.

Users who decide to participate in contact tracing download the app from the app store and register themselves. During sign-up they have the option to provide additional information on their identity such as their phone number, address, gender and age range (e.g. [20-30]). By voluntarily providing additional information to the CHA, and thus sacrificing on privacy, a user can enhance the effectiveness of the CHA, i.e. enhancing its power to study transmission dynamics, improving analytic capabilities and allowing for more effective preventive measures. For example, a phone number allows the CHA to actively call at-risk users, and by consequence ask for permission to subject them to a medical test. Acquiring data on the place of residence help the CHA to geographically map the spread of the disease. Thus, as required in *(P5) Voluntary participation*, users decide themselves upon participation, and the amount of personal attributes they share.

4.2 Local proximity sensing

Figure 2 illustrates how proximity (i.e. close contact) between two smartphone users is registered. Every user u generates an ephemeral public/private key pair, $Y_{u,t}/X_{u,t}$, for time period t, and locally broadcasts the public key $Y_{u,t}$. To avoid tracking, this public key is changed every N minutes. Upon physical encounters with other participating users, the public information is picked up by their devices. For each observed public key, each user calculates a shared ephemeral secret s_t according to a DH protocol. The user then computes a cryptographic hash from the concatenation of its own public key, the observed public key and the shared secret: *query_hash* $= H(Y_{own,t}, Y_{peer,t}, s_t)$. The obtained hash, *query_hash*, is used for query purposes only. A second hash for infection registration, *report_hash*, is obtained by interchanging the own public key and observed public key: *report_hash* $= H(Y_{peer,t}, Y_{own,t}, s_t)$. By using two separate hashes we limit the social graph building capabilities, as all query hashes are unique. The same holds for infection registration hashes.

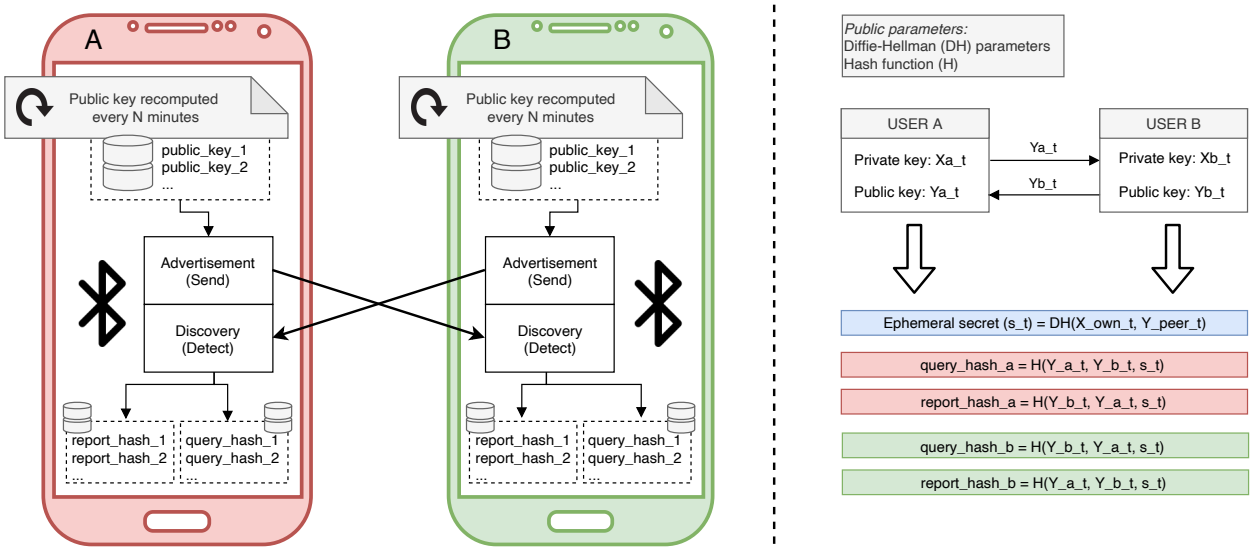

Figure 2: Close contact registration – two personal devices register their encounter by locally broadcasting their ephemeral public key and storing the resulting two locally-computed contact hashes. The public keys should be changed frequently to avoid tracking. On the right, a detailed description of the broadcast protocol at time *t*.

The following small example illustrates the protocol: user A and user B have a physical encounter. User A and B obtain each other's ephemeral public key, $Y_{a,t}$ and $Y_{b,t}$, respectively, and calculate the shared secret s_t as follows:

$$s_t = Y_{b,t}^{X_{a,t}} \bmod p = Y_{a,t}^{X_{b,t}} \bmod p \tag{1}$$

where $X_{a,t}$ and $X_{b,t}$ are the ephemeral private key of user A and user B, respectively. p is a public modulo parameter shared by all users. The public parameters are provided in certification kits which are obtained during app installation. For efficiency, in practice elliptic curve DH should be used to calculate s_t. After obtaining s_t, user A constructs their hashes:

$$query\_hash_A = H(Y_{a,t}, Y_{b,t}, s_t) \tag{2}$$

$$report\_hash_A = H(Y_{b,t}, Y_{a,t}, s_t) \tag{3}$$

where H is a cryptographic hash function. User B goes through the same procedure and constructs their hashes:

$$query\_hash_B = report\_hash_A = H(Y_{b,t}, Y_{a,t}, s_t) \tag{4}$$

$$report\_hash_B = query\_hash_A = H(Y_{a,t}, Y_{b,t}, s_t) \tag{5}$$

Note that the query hash of user A is the same as the report hash of user B and vice versa. Thus, it is possible to match them during risk score calculation.

Users store locally the hashes they compute for each of their encounter, together with the duration of the encounter and a distance estimate. The hashes are deleted after 14 days of registering, as this is the maximal incubation period, this satisfies *(F1c) Storage duration of 14 days*. After this period the hashes have lost their relevance. In addition, as mentioned above, users' ephemeral public keys change every N minutes. N depends on the ideal trade-off between functionality and privacy. Changing the public key more frequently prevents tracking, while changing the public key less frequently allows to determine the duration of the contact more accurately. As discussed in the functional requirements *(F1a)*, the ECDC labels an encounter of 15 minutes with an infected person as high risk. Thus, for optimal functionality $N = 30$ minutes. If a more frequently changing public key is required it is still possible to locally make an educated guess on what hashes correspond to an uninterrupted encounter with the same person.

We envision this protocol to be implemented on the Bluetooth BLE stack, and more specifically by leveraging beacon technology protocols [26]. BLE is local communication that allows for distance estimates thereby satisfying requirement *(F1b)*. BLE beacons normally do broadcast a UUID [27, 28], the protocol has to be adapted to not disclose this. We leave an exact implementation of this protocol open, as this depends on the technologies available on current personal devices, or even the efforts of a commercial partner designing a dedicated device.

In summary, our proximity sensing phase supports a realistic contact model *(F1)*. BLE power gives an accurate estimate on the distance between parties, while the rotation frequency of the public key ensures that contacts up to, let's say, 30

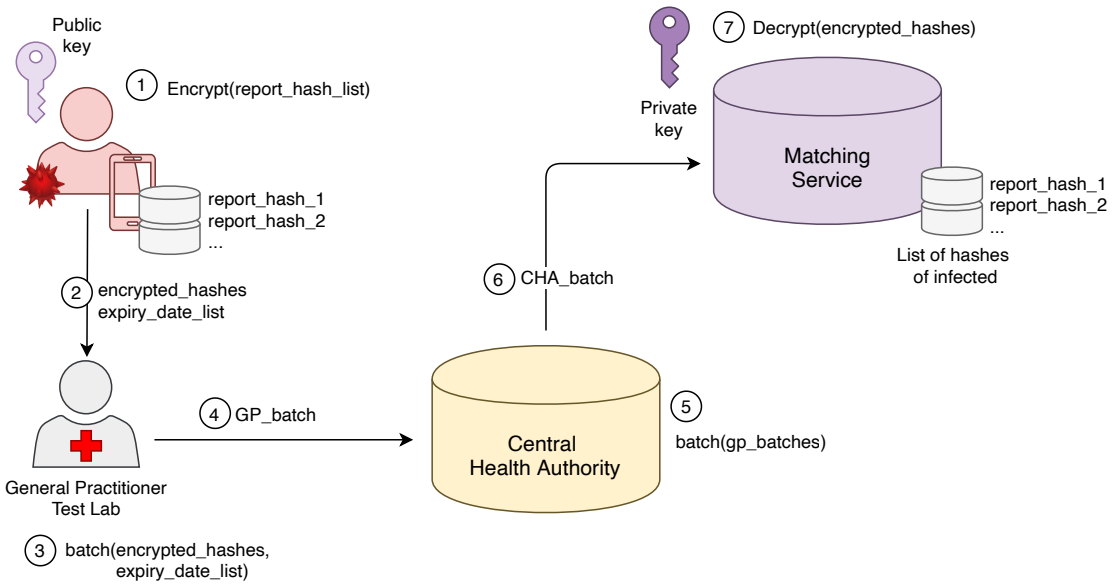

Figure 3: A new infection registration phase – a licensed general practitioner or test lab obtains the encrypted hashes of the person diagnosed with the disease and sends them in batches to the Central Health Authority (CHA). The CHA locally stores metadata regarding the certainty of the diagnosis, batches hashes from different users in batches and sends them to the matching service. The matching service stores the hashes. Alternatively, upon CHA request, a user can provide their register hashes to be included in the infected list.

minutes are stored. By eliminating non-close contacts, like contact that are shorter than 5 minutes at an intermittent estimated distance, our protocol ensures that the list of contact is complete *(F2)*, while achieving purpose limitation and data minimisation.

4.3 Infection registration

Infection registration can aid the study of transmission dynamics by sharing aggregated, anonymous information that are gathered by GPs and offered to the CHA. The protocol for infection registration is shown in Fig. 3. The goal is to build a database that contains all the hashes of the infected population. Special care is taken such that only the MS has access to this database and the MS cannot link hashes to the same user. Thereby we aim to satisfy *(P2) Infected user privacy*, as the hashes of infected users are never disclosed to the public; *(P1) Health status confidentiality*, as the MS does not know what hashes belong to which user; and *(P4) Interaction privacy*, as no entity learns what hashes belong to which user and with whose hashes they match.

Upon developing symptoms, a user goes to a GP or a test lab to get checked. If the user is tested positive, the medical expert demands the user's stored hashes for infection registration (*report_hash*), together with their coarse-grained expiry date. The expiry date is based on the day the contact happened. This information is required to obtain new insights w.r.t. transmission dynamics and to know when to remove the hash from the database. For privacy and security reasons, hashes cannot be stored longer than strictly necessary, i.e. 14 days as per *(F1c)*. The user encrypts their hashes with the public key of the MS and provides them to the GP through a local data transfer protocol. The GP batches and transmits the encrypted hashes of the infected users to the CHA. In a real system, this step could be integrated into manual contact tracing – a GP (or *health inspector* [29]) poses several questions to the diagnosed in order to identify close contacts that have been missed by the app and redact the reported hashes (e.g. based on specific time windows). The CHA batches the encrypted hashes over multiple users and forwards the batched data to the MS. Thus, the CHA and GP do not learn the hashes, and batching avoids that the MS learns what hashes belong to the same user. The MS then decrypts and stores the hashes. During health status query, it matches the hashes obtained in the query with the hashes in its database.

The CHA does not learn which of the registered users is infected. To enhance the information epidemiologists have to study – transmission dynamics, users can optionally indicate that they are infected in the application. To avoid linking user accounts with their report hashes, this notification is provided on a voluntary basis, and does not contain any link with the test that proved they are infected. For this extra information, we rely on the goodwill and honesty of users.

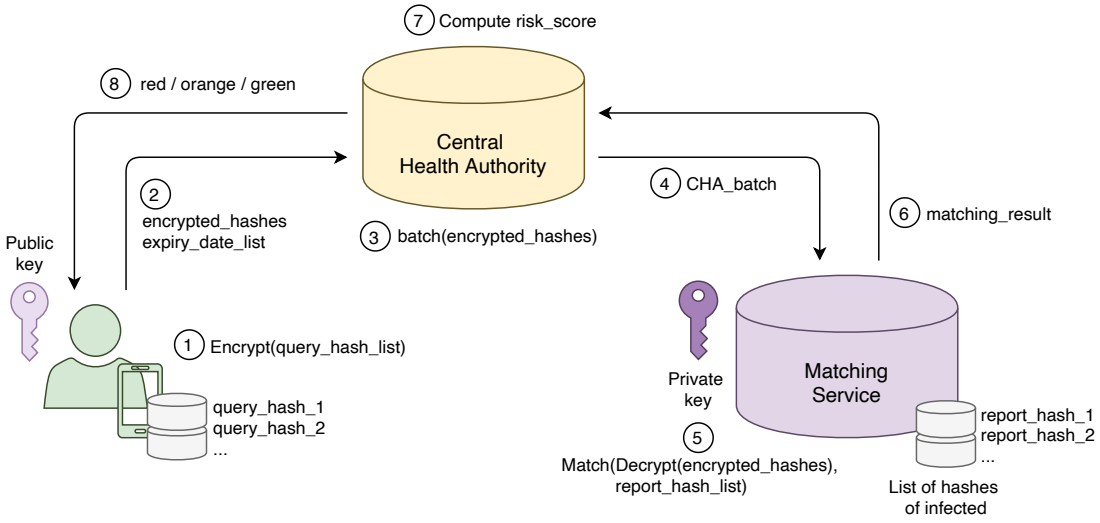

Figure 4: Health status query phase – the user sends their list of encrypted query hashes. The CHA forwards this list in batches to the matching service which decrypts the list and searches for a match within its database of hashes of infected people. The result is returned to the CHA which computes the risk score and informs the user of their infection status.

4.4 Health status query

Outsourcing matching to the MS (not the CHA) supports risk score computation that drives the mapping of the spread of the disease at CHA *(F4)* and enables user notifications *(F3)* while protecting user privacy *(P1)* and *(P4)*. Upon user request or at a fixed time, the user's health status is determined. The CHA computes the risk score according to the actual transmission dynamics model and notifies the user. The user will not learn the exact risk score, but a colour encoded result that translates their risk score into an advice, this satisfies *(F3) Notifying at-risk individuals*. Green corresponds to very low chance of infection, and the user is free to travel. Orange implies suspicious of infection, so the user is advised to only make strictly necessary translocations. Red means high chance of infection, self-quarantine is mandatory and tested is encouraged. To compute the exact risk score, the CHA asks the MS to indicate what encrypted query hashes match with hashes in the infected list. The CHA never learns the hashes, while the MS is not able to link hashes from the same user, thus complying with *(P4) Interaction privacy* and *(P1) Health status confidentiality*. We trust the CHA and the MS to act honestly when respectively notifying the user and performing the matching, therefore our protocol supports *(S3) Notification integrity* and *(S2) Contact integrity*. The latter relies on the use of the unique per-user-per-contact identifiers as explained in Section 4.2.

Figure 4 shows the interactions during health status query phase. The user encrypts their stored contact hashes for querying (*query_hash*) with the public key of the MS and sends them with some metadata to the CHA. The CHA batches the encrypted hashes over different users and queries them to the MS, while keeping the metadata for itself. The metadata contains at least the duration and proximity of the contact, and how long ago the contact happened. No location information is stored or processed, thereby satisfying *(P3) User location privacy*. The MS decrypts the received hashes and compares them to its database of reported hashes of infected people, *report_hash*. The MS indicates to the CHA what encrypted hashes matched with the infected list. From the metadata, the CHA estimates the probability that this user is infected according to the latest transmission dynamics model – thereby supporting changing transmission models as per *(F1)*. The resulting risk score is translated in a colour which is send to the user. The CHA keeps track of the risk scores of the population – fulfilling *(F4)* – to comply with its goals, i.e. study transmission dynamics and map the spread of the disease.

Communication cost optimisation. In our current scheme, we encrypt every hash separately with the public key of the MS. If we set our protocol to use SHA256 as a hash function (hashes with a length of 256 bits), RSAES-OAEP as an encryption/decryption algorithm with a key size equal to 2048 bits, and assume that on average each user will query 200 contacts (i.e. 200 hashes) per day, then the communication cost as a result of the encrypted hashes would be ≈ 51KB (200 x 2048). To reduce the computational and communication burden on the user's device due to public key encryption, we apply the same optimisation as during infection registration: a batch of n query hashes (e.g. $n = 50$) is encrypted with a symmetric encryption scheme, such as AES. Thus, in the case of a batch of 50 hashes and use of symmetric encryption key with a length of 256 bits, then the communication cost as a result of the encrypted hashes would be

≈ 2.63KB (4 x (50 x 256 + 2048)), which means an approximate of a 20-fold reduction in communication cost. This implies that the public key of MS will only be used for transporting symmetric keys from the users (both querying and infected) to MS. However, in practice, one can instead use a key encapsulation scheme, which is more suitable for transporting keys than a public key encryption scheme. This scheme is also used during infection registration. As this batching allows the MS to link n hashes, i.e. $n = 50$, the amount of hashes batched together is effectively a trade-off between privacy and communication (and computation) overhead.

Scalability and interoperability. Our architecture is practical, in the sense that it is easy to implement, as it can be built with simple existing crypto building blocks and builds on already existing technology stacks. Furthermore, the system can be rolled out in the whole of Europe as the account created with the CHA can rely on the European eIDAS [30] technology which allows for remote identity verification of all European identity documents. Thus, when a user travels to a foreign European country the user application could simply create an account and maintain contact with the CHA of the foreign country. This and the above ensure that our protocol fulfills *(F5) Work across borders, scalable and realisable the with current technology stack.*

5 Security and privacy analysis

Due to the sensitive nature of COVID-19 and contact tracing, flaws in the design of such a solution could have disastrous consequences. In this section, we describe the different threat actors by describing the adversary's goals, capabilities and knowledge. This is followed by our security and privacy analysis.

5.1 Threat actors

We enumerate the potential threat actors used in our security and privacy analysis. Every attacker is described by their knowledge, capabilities and goals.

User. An honest-but-curious user aims to de-anonymize their encounters and obtain sensitive information about them. As summarised in Table 1, they can target their encounters' health status, PII, trajectory or encounters of other contacts. This type of users is capable of querying the system to learn the health status of their contacts. Furthermore, they can collect a manifold of encounters by strolling through the city. They can collect additional meta-data on their contacts by storing the exact time and location of encounters (e.g. installing another app that leverages the same communication infrastructure).

Malicious user. A malicious user has a double aim: first, they try to obtain sensitive information on their encounters; second, they seek personal benefit by tinkering with risk scores. For example, they could force the authorities to test them by raising their risk score; or they could raise a target user's risk score in the hope that they would self-isolate. Malicious adversaries are also technically more skilled then the honest-but-curious users. They can inject arbitrary contacts by directly modifying the app, and register multiple accounts. They can also use antennas (e.g. owned) to eavesdrop local communications, inferring public material broadcasted by other devices. Additionally, as they are not honest they can falsely claim that they are infected.

GPs. GPs are an honest-but-curious threat actor. They follow the protocol specification, but might want to know additional sensitive information about their patients. GPs already know their patients' identity and health status, however, they might also want to learn their patients' trajectories and encounters. We trust GPs to act with integrity and thus diagnose patients truthfully. With other words, they will not knowingly register fake contact hashes as infected when they are actually not. GPs who do not adhere to the aforementioned break their professional integrity and will be held accountable.

Central health authority. The CHA is an honest-but-curious party that coordinates the protocol. Apart from learning the outcome of the protocol (the health status of the population), they might also aim to gain out of scope knowledge on the population – a social interaction graph and trajectories of individuals. They might try to obtain this new knowledge by aggregating information from other data sources it normally has access to. We assume that this threat actor will not turn malicious and undermine the functioning of the underlying system, e.g. send fake notifications or enforce policies on arbitrary users. This will clash with its goal of curbing the spread of the disease.

Matching service. The MS acts in an honest-but-curious manner and aims to gain out-of-scope knowledge. It is regarded as an external entity in Table 1 and thus tries to de-anonymise users, learn about their health status, construct their trajectories or build social interaction graphs. To this end, the MS can use its knowledge on hashes used for health

status queries and the hashes in the infected list. Thus, it tries to link hashes to users on a large scale. We trust the MS to perform the matching operation honestly. Moreover, the MS will not claim false matches or withhold real matches. If the MS does attempt such feat the company hosting this service will be held accountable.

Malicious third-party. A malicious third-party extends the malicious user. Compared to the malicious user, they additionally have the goal to disrupt the service (e.g. DDoS) and cause panic. Furthermore, its capabilities are scaled up. Thus, they can obtain a manifold of accounts and can roll out a network of antennas and Bluetooth transmitters at scale.

Malicious matching service. A malicious MS has the same goal and knowledge as the honest-but-curious MS but has the capabilities of the malicious third party. Thus, they know the list of infected hashes, query hashes and can install antennas at scale.

Colluding MS and CHA (and GP). A colluding MS and CHA can link query hashes to users. If GPs also participate, registration hashes can be linked together as well. The likelihood of GPs colluding depends on what legal entity plays the role of CHA, and what the GPs legal obligations towards a health authority is. If these entities collude we should also assume they installed a large scale eavesdropping infrastructure. Thus, this threat actor extends the malicious matching service with the ability to link users to hashes and de-anonymise the user. The aim of this threat actor is to build a social interaction graph and to track users.

5.2 Security analysis

(S1) Infection integrity. Our protocol defends against malicious users falsely claiming infection (aiming to cause panic) by limiting who and when can report such infections. In our solution, only authorised medical experts (including GPs) are allowed to report infections; users cannot report such infections directly to the system, instead, they will have to contact their GPs first. In addition, GPs report infections only after suspected users (with symptoms or at high risk) have been proven to be infected by means of testing. Hence, our solution supports (almost) zero false positive rate.

(S2) Contact integrity. Our protocol ensures that only real physical encounters impact users' risk score and analytics about transmission dynamics and the spread of the disease. In other words, attackers cannot inject false encounters to users' lists of close contacts which would affect users' risk score as well as analytics on the disease. This is achieved by combining several defence mechanisms. First, publicly broadcasted information (users' ephemeral public keys) are not used directly for reporting infections nor querying for health status, instead locally computed hashes are used. Second, each encounter is registered as hashes that, apart from users' broadcasted ephemeral public keys, take as input also a shared secret known only to the two users in close contact. Third, each encounter is registered by both users using two different hashes in reverse order – one for registering infections and one for querying health status. These measures ensure that our system supports unique per-user-per-contact identifiers – hashes – which are not broadcasted to the public and which can be computed only by the two users in close contact (due to the use of a shared secret key as an input for the computation). This makes injecting false contacts with any impact extremely unlikely.

In addition, our solution protects against replay attacks of observed public keys resulting in a cloned contact due to the shared secret between close contacts that is used as a salt for computing hashes. An attacker - the malicious user, malicious MS or malicious third party - approaching their victim will now observe a public key that is insufficient to impersonate them during future broadcasting sessions.

Furthermore, an adversary - the malicious user, malicious MS or malicious third party - might attempt to register clone contacts by launching relay attacks – relaying communication between two regular users at different locations. Assuming one of the two is at high risk of being infected, this attack can trigger false positives for a targeted subset of the users. However, as pointed out by [31], this attack could be prevented by incorporating information about the absolute location of the user or using side-channel measurements – such as the vibrations from the surrounding environment [32]. Given the low impact and high-cost of performing this kind of attack on our solution – it has to be performed in real time for the involved parties to share a secret – the discussion on mitigations is beyond the scope of this paper.

(S3) Notification integrity. As our solution ensures infection integrity and contact integrity (using the same reasoning as above), and assuming that the MS and the CHA honestly perform the matching and calculate the risk score, respectively, a user will receive a notification that they is at risk only if they had close encounters with infected users. Note that such assumptions – central entities calculating risk scores correctly and/or broadcasting the correct information regarding infected users for local risk score calculation – are inevitable for all centralised and decentralised solutions.

5.3 Privacy analysis

(P1) Health status confidentiality. In our solution, the health status of users is known only by the health authorities (including the GPs), as they compute the risk score of the individuals, and by the users who are notified of their own risk score by the health authorities. As the MS receives only the hashes of infected users, and these hashes do not contain any PII of users, there is no way for an honest-but-curious MS to identify the infected users. Moreover, as these hashes are provided to the MS in batches (batched first by GPs and then by the CHA), the MS does not even know which hashes belong to the same user. Similarly, when users query the CHA for their health status, these hashes are again batched at the CHA before being sent to the MS to perform the matching. Hence, although it learns the number of matches per each CHA query, the MS does not know to whom these hashes (and matches) belong to, nor the number of the users (and their contacts) whose health status was queried. Note that, if users deploy an optimisation technique – instead of encrypting every single hash with the public key of the MS, they encrypt batches of n hashes (e.g. $n = 50$) with a symmetric key and then use the public key to transport this symmetric key – the MS would be able to link n hashes to the same user.

However, if the MS turns malicious and it is able to get in close contact with possibly infected users, it will be able to de-anonymise them once it sees their hashes reported. This threat holds true, although it is slightly harder to carry out, for every user who is able to be in close contacts with their target, i.e. honest-but-curious users, malicious users and malicious third-parties. They can query the CHA multiple times with only a single contact in their list. Eventually, the coarse-grained score returned by the system will reveal some information about the victim. It is worth to notice that no contact tracing solution protects against the accidental disclosure of health information. If a regular user encounters only, let's say, three people in the past two weeks and gets notified to be at-risk, they will probably be able to de-anonymise the infected user by remembering a small group of individuals. This is inherent to every contact tracing solution.

To limit the impact of these possible attacks, the system could deploy the following mitigation strategies. Honest-but-curious users can be prevented from learning the status of their close contacts by simply enforcing a limit on the number of queries associated to each account. To protect against the stronger actors, that can register multiple accounts, the CHA can require the authentication of the users by means of a permanent or semi-permanent ID (see Section 3.4). This exposes users' privacy since the CHA will be able to associate risk scores to identities instead of anonymous accounts. However, this is instrumental to rate limit the amount of queries performed by an entity capable of registering multiple accounts. Therefore, this enormously reduces the possibility to disclose the health status of infected and at-risk users to third-parties.

(P2) Infected user privacy. As the risk score is calculated by CHA based on the matching results provided by the MS, users never get to know which of their close contacts affected their risk score being high. Moreover, as the CHA never gets to see the hashes of contacts, the CHA also does not know this information. As already mentioned above, the MS is the only party that has access to the hashes of users and does the matching, it will know exactly which two hashes matched. However, as the MS does not know to whom these hashes belong nor the metadata such as proximity and duration of the contact, it cannot compute risk scores of users who query nor link them to the real identity of users.

(P3) User location privacy. Our solution never collects, stores or processes any location data. By frequently changing the advertised public key, we limit the possibility to track the same user within a very strict time window (max. 30 minutes). In addition, our solution also protects against malicious and powerful MS who runs network of beacons and registers close encounters with all users. Such a powerful adversary will have access to query hashes for all users (including the location and time when it registered them) as well as it would have managed to insert the corresponding report hashes in the users' lists. As they also receive all the register hashes of infected users as well as query hashes of the users who query as part of our protocol, the MS could potentially link every hash it receives to a location and timing. However, as our solution always provides the hashes to the MS in a shuffled and batched form, the MS does not learn which hashes link to the same user, hence it cannot construct users' trajectory.

Of course, if the GPs, CHA and the MS (who runs a network of beacons) collude, then they can track users' past movements by linking the known hashes. To mitigate against this (unlikely and unrealistic) threat, our system could be enhanced with the solution proposed by [5], i.e. sharing the public key in several pieces ensuring that at least N contacts have been made before a user can determine the public key. While being computationally inexpensive, this further reduces the tracking capabilities to a very short time window of less than 1 minute. Additionally, a colluding CHA would have to install a capillary network of transmitters to obtain enough packets.

(P4) Interactions privacy. Our solution protects users' interactions privacy by not allowing social graphs (both global and local proximity interaction graphs) to be built by the individual threat actors. In terms of attempts by external adversaries to build a global interaction graph, in our solution, none of the hashes are ever broadcasted in the public, which prevents adversaries from discovering contacts between users, i.e. build a global interaction graph. This is due to

the decoupling between the query hash and the report hash. Even by knowing the public keys of target users, it is not possible for the adversary to derive their private keys and computationally hard to build a rainbow table that contains all the possible salted hashes (assuming that the DH parameters are selected properly), as explained in Section 4.2.

In addition, none of the GPs and the CHA have access to any of the user hashes as they are encrypted at the users' devices and decrypted only when they reach the MS. This prevents the GPs and the CHA of building any local proximity graphs of the users who upload their report hashes or query their health status. Even though the GPs know the identity of the infected users and the CHA might know the identity of the users who query, they both do not have access to the users' hashes, hence cannot build any kind of local proximity graph. As the MS is the only entity that has access to users' hashes, our solution provides additional protection measures against local proximity interaction graphs building: shuffling and batching the encrypted hashes by the GPs and the CHA, so that the MS cannot link the decrypted hashes belonging to the same user. More specifically, for infected people, a double step of batching is performed: first, the GPs batch encrypted hashes before sending them to the CHA; second, the CHA concatenates the encrypted hashes before providing them to the MS. In this way, apart from the CHA not learning which account is associated to a specific set of encrypted hashes, the MS cannot link infected hashes together. For people who query the CHA, the contact hashes are encrypted locally and batched at the CHA before being sent to the MS. The CHA is precluded from learning the interactions between an at-risk user and their close contacts, while the MS cannot link multiple hashes to one specific user, therefore learning no useful information from matching hashes. However, as mentioned earlier, if users deploy an optimisation technique – encrypt batches of n hashes (e.g. $n = 50$) with a symmetric key – the MS would be able to link n hashes to the same user.

Due to the use of unique per-user-per-contact hashes, the MS can link queries of the same user if the user sends in each query the full list of their past encounters. Therefore, to mitigate against such fingerprinting, the user can only send an update of their contact hashes, i.e. the contact hashes they acquired since their last query. When the list of infected users is expanded, the MS needs to match them against past contact hashes as well. Therefore, the MS remembers the contact lists of each query for 14 days and rechecks for matches when new hashes are added to the infected list. It sends the updates to the CHA together with a unique identifier for each contact list, e.g. the hash of the list. The CHA remembers who the contact list belonged to, recalculates and fuses the risk scores of every query of that user.

However, if the MS, GPs and the CHA collude, they can construct a proximity interaction graph of the infected and at-risk users. Thus, the MS and the CHA must be two different legal entities, with competing interests. Recent work has proposed a charter for the selection of trusted intermediaries among existing governmental institutions [33]: the MS could be provided by private actors, while the role of CHA is filled in by the government. One way to prevent this privacy nightmare is by leveraging trusted execution on the MS. By doing so, all the computations happen in a trusted enclave, the MS never learns the infected list nor the contact list. We leave such an extension to TEE experts. A possible extension, computationally expensive but feasible in a centralised scenario, leverages private set intersection [34]. We can reduce the degree of trust we put on the authorities by only granting them access to the intersection between the infected list and the query list instead of sharing the entire list of query hashes.

6 Centralisation vs. decentralisation

It is not always clear whether the advantages of a centralised solution outweigh its disadvantages. In this section, we outline the main differences between computing the risk score in a central location and deriving the score on the user's device. Next we compare our design to two solutions: (1) a centralised solution, i.e. ROBERT-alike [23], in which the diagnosed patient uploads a list of the observed identifiers and the user contacts the server to obtain their health status; (2) a decentralised solution, i.e. DP-3T-alike [5], in which the diagnosed user uploads a list of their own broadcasted identifiers that is made available to the public by a back-end server. We delve into each solution by analyzing their functionalities, and their security and privacy requirements. Table 2 presents a summary of our comparison.

Risk score function update. Decentralised solutions do not allow health authorities to compute a risk score for the users. Users can come up with their own risk score function based on the local information and the information they receive from a central server. This does not allow to tailor the response of the system, unless a user directly contacts the health authorities or provides a digital clinical diary to the central server (analyzed by epidemiologists). Some of these information can be deemed instrumental to combat the disease, but we must consider (1) the large adoption needed for the app, i.e. around 56% of the population or 80% of the smartphone owners in the UK [2], and (2) the number of people willing to integrate new information on a voluntary basis, which is only a fraction of the userbase. This might lead to insufficient data to perform a meaningful statistical analysis. A centralised approach allows to tailor the score function to several factors that are voluntarily provided by the users. Based on these variables – such as coarse demographics or recent symptoms – the server can compute a more accurate risk score for the user, based on professional consultation while achieving a better characterisation of the disease, which in turn leads to an improved

Table 2: Summary of decentralised vs. centralised solutions. We use four colours that identify the severity and feasibility of each threat in each scenario: green (low impact), yellow (medium-to-low impact), orange (medium-to-high impact), and red (high impact). For functionalities, the colours only represent to which extent a specific functionality is present.

	Requirements	Decentralised	Ours	Centralised
Functionality	*Risk score function update*	Yes (partial)	Yes	Yes
	Risk score root of trust	No (possible)	Yes	Yes
	Transmission dynamics	Voluntary	Yes	Yes
	Further analytics	No	Yes	Yes
Security	*Infection integrity*	Yes	Yes	Yes
	Contact integrity	Own-injection, replay & relay attacks	Only relay attacks*	Own-injection, replay & relay attacks
	Notification integrity	Public DB	Targeted attacks	Targeted attacks
Privacy	*Health status confidentiality*	Rightful user	CHA (anonymised)	CHA (linkable)
	Infected user privacy	Close contacts & GP	Close contacts (limited) & GP†	Close contacts (very limited) & GP
	User location privacy	Malicious users & third-parties (infected users)	CHA (very limited)‡	CHA (central IDs)
	Interactions privacy	Preserved	CHA (proximity)¶	CHA (proximity)

* Our solution requires physical proximity, hence a user cannot fake a contact with another user just by knowing their public key. However, if a malicious user comes in contact with a rightful user, the unique contact can be uploaded by a third user soon-to-be diagnosed positive and colluding with the malicious one. This attack is highly unlikely and difficult to scale in practice since it requires.

† The user learns their health status while being limited in the amount of queries they can perform (more powerful attackers are countered via authentication). This makes de-anonymisation very hard for close contacts. If all GPs collude, CHA could de-anonymise the infected users.

‡ If CHA and MS collude, this might lead to location tracking and de-anonymisation of the users. The colluding parties would need the decrypted unique hashes uploaded by the users at query time and a network of Bluetooth enabled-devices to fake close contacts with the victims (e.g. less than 2m for more than 15min).

¶ In case of collusion with all GPs and MS, CHA can build a proximity graph of the infected users and their close contacts. This is easier for the centralised solution, in which the pseudonyms are generated centrally and a timestamp is associated to each contact.

risk score function. This is especially useful since we lack a thorough understating of some major factors, like the viral load and the infectious dose [35]. Hence, a central approach allows a central server to proactively inform the user of a change in the policies that results in a different health code. This additionally helps to study the transmission dynamics model. By consenting the central authority to know their risk score, the user will not need to perform a query or update the app to be notified. In our solution, meaningful statistics are derived without the need for identifying the user, by only exploiting the risk score and non-identifiable, coarse-grained metadata. This is in line with *data minimisation*, since mapping the risk score of the users might be instrumental to fighting the disease. In our protocol, the user can voluntarily provide contact information for specific needs – such as establishing a direct communication channel for testing or requesting a consultation.

Risk score root of trust. As the authors of ROBERT [23] point out, to maximise the impact of contact tracing, at-risk individuals that did not develop symptoms yet should be tested. This allows to break the transmission chain earlier compared to solely encouraging at-risk individuals to self-quarantine. The chain is broken early by testing asymptomatic at-risk individuals because these users could have infected others before learning about their at-risk status. If they would only self-isolate, the chain is broken in a less effective manner as their past encounters will not be warned they are at-risk until they develop symptoms themselves, which could be several days later. Therefore, a trustworthy source that can attest the risk score is required. In decentralised systems, this is harder to obtain as risk score calculation is performed locally. Only by the use of trusted execution this could be reached. In a centralised system, proof of the score can be issued by the central party. A GP can use this proof to determine upon testing. Such an extension is not described in detail, but is very similar to the proof issued by the GP for infection registration in ROBERT [23].

Transmission dynamics: data available to epidemiologists. Epidemiologists that study the transmission conditions of the disease require to know about the nature of contacts within the population. The nature of the contact is determined by information on the interaction itself, and the health status of the interacting parties. All solutions use similar information to characterise the type of the contact, i.e. duration, proximity, date, etc. The information on the health

status of the interacting parties is different. In the decentralised solution the contact meta-data of contacts with infected individuals is shared, in aggregated form, by at-risk users only. Thus, epidemiologists can study the contacts of healthy at-risk and sick at-risk individuals. The users can indicate voluntarily whether a medical test was performed and what the result was. In the centralised solution, the meta-data on the contacts is shared with the system infection registration or health status query. In our referenced centralised solution, the meta-data is shared during infection registration, this implies that contact information between infected and healthy individuals is known to epidemiologists. As the system learns what users are infected and knows their ephemeral IDs, the reference centralised solution can also provide insights on the contacts between infected people. However, they have no way to provide information on the contacts between healthy users. Our system provides meta-data on the contacts during health status query, therefore the system knows about the meta-data of contacts between healthy individuals – i.e. the query hashes without a match. The contact between healthy and infected people can be studied when a match with the infected list occurs. To characterise the contacts between infected individuals, the system can rely on users indicating that they have been infected, like they do in decentralised systems. Voluntary systems, however, rely on the cooperation of the population, as discussed earlier. Thus, our system can provide the most complete information w.r.t. the nature of infections, however, we rely on voluntary participation. The reference centralised solution on the other hand only misses the meta-data on contacts between healthy people which, at first sight, is the least important.

Further analytics: mapping the spread of the disease. In centralised systems, the central server learns the risk score of the population. This allows it to track the spread of the disease. A more precise geographical map can be computed based on the voluntary information provided by the users. In the decentralised systems, the users optionally share their risk scores and only when their score is high. This does not allow to get any in-depth insights on the degree of transmission among the population.

(S1) Infection integrity. All the solutions support infection integrity by only allowing medical personnel (e.g. GPs) to grant the upload of the ephemeral IDs.

(S2) Contact integrity. There is no solution ensuring the integrity of every contact. Decentralised and centralised solutions offer no protection against the injection of a publicly known pseudonym in a user's own list. This might become problematic for two reasons: (1) the government wants to scale testing but there are no guarantees about a self-declaration of close contact, (2) the private companies – especially during a less strict phase of lockdown – have to agree on letting one employee to work from home and self-isolate. For example, in a decentralised solution, a malicious user who wants to be tested to be reassured about their health status might inject a publicly available infected pseudonym in their mobile device. They can later ask to be tested by reporting them to the authorities and showing the app notification. Alternatively, in centralised solutions where infected users claim who they have been in contact with and with what intensity, there is the threat of that user manipulating someone else's risk score. The user who the infected user interacted with has no way to refute that the interaction happened or to claim that the nature of the interaction was different. In our solution, a directed attack like this is impossible as both parties need to acknowledge the contact and the meta-data is provided by the user querying.

Both decentralised and centralised solutions suffer from replay and relay attacks. This is due to the impossibility to verify whether an attacker has re-broadcasted a pseudonym they has previously observed from a rightful user. In a replay attack scenario, this leads to many false positives in case the target user becomes sick, therefore threatening the integrity of the system. A recent work proposes a mitigation for our reference decentralised solution based on a radically different protocol [31]: by allowing two users to interact, they can compute a message authentication code (MAC) based on a challenge and a timestamp. While working within a limited time window, this mitigation does not fit in current schemes that assume no communication between two users. The centralised solution proposes the use of a timestamp and a MAC to minimise the window of opportunity for a replay attack to a few seconds. We argue that short time windows only do not prevent replay attacks, while the use of timestamps opens new attack surfaces to infer users' interactions. In a relay attack scenario, there are no guarantees about the relayed ephemeral ID. This means that relying, in real-time, the ephemeral ID that will be reported by the diagnosed patient is enough to generate a fake notification.

Our solution relies on unique per-user-per-contact hashes and obfuscates these hashes to the public, thereby reducing the possibility to cheat and increasing the probability that testing close contacts will not result in a waste of resources. It offers a protection based on the generation of a shared secret that is unique for a contact. This ensure stronger protection than, for example, including a timestamp. As discussed in Sec. 5.2, given the bi-directional nature of our protocol, the success of replay and relay attacks is severely impaired. In particular, a third-party cannot relay public identifiers indiscriminately and must target one pair of users within the refresh window of their public keys. If only one of the two public keys is transmitted and the owner of the public key is tested positive, the receiving party would not be notified of being at risk.

(S3) Notification integrity. All the solutions rely on an honest back-end server that manages pseudonyms. However, in decentralised schemes, the database is publicly available, therefore auditable by third-parties. Assuming that the system decides to tweak its records, this would result in an indiscriminate attack that would undermine the functionality of the system itself with little to no effect on end-users. Centralised solutions, on the other hand, can target specific users by assigning them a risk score that is based on out-of-scope information. This can have severe consequences – such as discrimination against minorities and the poorest. In our base solution, we do not link the identity of the user to their account and distribute the information based on the role of each entity. This prevents function creep by a malicious third-party in charge of coordinating the protocol. However, we still rely on a (trustworthy) CHA to behave correctly. An honest-but-curious CHA might still target users for the sake of achieving its goal of halting the disease from spreading, ending up causing harm.

(P1) Health status confidentiality. Due to the local broadcasting of pseudonyms, de-anonymisation is possible in decentralised schemes, specially for malicious users and malicious third-parties. They can use a separate app that sniffs the pseudonyms and stores them with the exact location and time of the encounter. Centralised solutions provide the health status to the coordinator of the protocol but hide it from the public. This protects against malicious users and third-parties but requires a substantial degree of trust in the health authority. In our solution, the list of (unlinkable) hashes of infected users is only learned by the MS, never by the general public. The risk score is never revealed to the users while only a three level colour encoded feedback is provided. By enabling strong authentication, we can additionally defend against reconnaissance attacks by rate limiting the amount of queries users can make. The risk of data breaches remains like in centralised solutions, but we minimise its output. As described above, a collusion between CHA and MS makes infected users and their close contacts potentially de-anonymisable.

(P2) Infected user privacy. Follows from (P1). It is easier for regular users of a decentralised scheme to obtain information about their close encounters. In our reference centralised solution [23], the user does not share their contacts pseudonyms, hence it is harder for a malicious user to obtain information about their encounters but still feasible by means of multiple accounts. Our solution protects against regular users by rate limiting queries. Only powerful adversaries can obtain a coarse-information (colour encoded) about the health status of their close encounters (the ephemeral public key is not enough). This threat is mitigated by adding strong authentication, as discussed above.

(P3) User location privacy. Decentralised solutions can be susceptible to location tracking of infected people if the pseudonyms of the infected people are uploaded or leaked. Solutions that register observed identifiers, instead of the ephemeral IDs generated by the infected, preserve the location of people who tested positive while revealing meaningful connection between users, i.e. co-location. This is due to the use of the same ephemeral ID in the phases of broadcasting and uploading, which remains valid within a specific time window. The centralised solution is prone to mass surveillance by means of de-anonymisation and location tracking. In this solution, the observed pseudonyms are generated centrally, therefore they can trace back movements by assigning carefully crafted semi-permanent IDs to the users of the system. This only requires a network of Bluetooth enabled devices and observing the shared IDs. Additionally, timestamps are used that make linkage attacks easier to carry out. Despite being susceptible to location tracking, our design is inherently safer: a user queries by uploading only hashes generated from close contacts, i.e. the colluding CHA will need to generate several close contacts (e.g. less than 2 meters for more than 15 minutes) to build a meaningful history of the locations of a specific user. Moreover, the hashes are never broadcasted nor linkable to the public key of the user, and unknown to CHA.

(P4) Interactions privacy. In decentralised solutions, a capable third-party can only build a limited interaction graph within the pseudonym refresh time window. However, this is only possible if the observed identifiers are shared rather than infected ones (different from our reference centralised solution). By generating the ephemeral IDs centrally, CHA can build a proximity interaction graph by simply observing the pairs (ID, timestamp) uploaded at the time of registration of a new infection. Even by generating the IDs on the user's device, it is easy for CHA to learn co-location: if two diagnosed users, A and B, have been in close contact with a third (diagnosed) user, C, within a certain time window, A and B will report the same ID shared by C. Our solution ensures similar protection to the decentralised solution by computing a unique hash per contact and by using different hashes for querying and reporting. CHA does not learn that a contact happened between two non-infected users and cannot link multiple hashes based on timestamps. This prevents CHA from learning co-locations, i.e. whether two users have been in close contact with a third user, due to the use of different hashes. The link between the user who performs the upload and its contacts hashes can be further weakened by using proxies, as outlined in our reference centralised solution [23]. In their solution, however, the contacts remain linkable via timestamps.

Summary. Decentralised and centralised solutions have several strengths and shortcomings. On the one hand, decentralised solutions that rely on the integrity of medical reporting do not allow to proactively refine policies

(e.g. allocate more resources to specific areas) or to directly contact potentially infected users. These solutions expose the privacy of infected people by allowing users to check for infections within their close contacts list and trust the users to behave correctly, hoping for them to disclose key data to characterize the transmission of the virus. On the other hand, centralised solutions expose users' privacy to several threats: data repurposing during and after the epidemic, linkage attacks and de-anonymisation, state-enabled surveillance, and data loss in case of breaches. Hence, a high degree of trust in the central authority is instrumental to a successful tracing. We envision a central authority that only knows one important attribute about users – i.e. their risk score. Other variables – such as the precise absolute location and the list of close contacts – are not essential to halt the spreading of the virus and enable function creep by the authorities in charge of coordinating the protocol. In centralised solutions, the risk score computation can be aided by epidemiologists, which can decide to modify this function based on novel discoveries. This can be problematic in more decentralised ones (e.g. requires updating the app). By spreading trust amongst entities, our solution strives to achieve the best of both worlds, minimising the privacy risks for the users while allowing the central authority to fight the disease by leveraging the appropriate tools. We trust the CHA to behave honestly and the MS to carry out the matching on behalf of the CHA while limiting the consequences of collusion between the two. A privacy-preserving solution reduces the amount of trust a user has to put in the coordinator of the protocol. This might enable wide adoption, which is a fundamental factor in a successful digital contact tracing campaign.

To conclude, we stress that contact tracing needs a strategy encompassing several factors. Proximity-based exposure alert is prone to false positives, hindered by a low adoption rate, and a harbinger of discrimination and false sense of security [36]. Nonetheless, technology can be of great help. We advocate for the development of an app to go in conjunction with a wide-testing policy. Digital contact tracing helps to scale up the job currently undertaken by human interviewers but does not replace a holistic approach involving public institutions across different sectors.

7 Conclusion

In this paper, we have proposed a privacy-friendly yet practical and effective contract tracing solution. To strike the balance between functionality and privacy, our solution combines local proximity tracing via unique per-user-per-contact hashes for user privacy protection and centralised risk score calculation for increased functionality in terms of data analytics support for health authorities. Furthermore, for additional protection against de-anonymisation attacks, infected users' hashes are never revealed to health authorities nor the general public. In addition, our solution is extremely simple and practical as it only relies on public key infrastructure. We believe that its advanced functionality, sufficient privacy protection and simplicity make it as effective as fully centralised solutions while respecting users' privacy. We hope that the solution will be quickly adopted and used in practice to combat highly-contagious diseases.

Acknowledgement

This work was supported by: CyberSecurity Research Flanders (Dutch: Vlaams Impulsprogramma voor Cybersecurity) with reference number VR20192203, and imec through the Security & Privacy Centre projects on Secure Distance Bounding. This research is partially funded by the Research Fund KU Leuven. Mustafa A. Mustafa is funded by the Dame Kathleen Ollerenshaw Fellowship awarded by The University of Manchester.

References

[1] Covid-19 basics. `https://www.health.harvard.edu/diseases-and-conditions/covid-19-basics`. Accessed: 07-04-2020.

[2] Luca Ferretti, Chris Wymant, Michelle Kendall, Lele Zhao, Anel Nurtay, Lucie Abeler-Dörner, Michael Parker, David Bonsall, and Christophe Fraser. Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing. *Science*, 2020.

[3] EDPB letter concerning the european commission's draft guidance on apps supporting the fight against the COVID-19 pandemic. `https://edpb.europa.eu/our-work-tools/our-documents/letters/edpb-letter-concerning-european-commissions-draft-guidance-apps_en`. Accessed: 17-04-2020.

[4] Ran Canetti, Ari Trachtenberg, and Mayank Varia. Anonymous collocation discovery: Taming the coronavirus while preserving privacy. *arXiv e-prints*, page arXiv:2003.13670, March 2020.

[5] Decentralized privacy-preserving proximity tracing (dp-3t). `https://github.com/DP-3T/documents`. Accessed: 07-04-2020.

[6] Covid watch. `https://www.covid-watch.org/article`. Accessed: 07-04-2020.

[7] Ramesh Raskar, Isabel Schunemann, Rachel Barbar, Kristen Vilcans, Jim Gray, Praneeth Vepakomma, Suraj Kapa, Andrea Nuzzo, Rajiv Gupta, Alex Berke, et al. Apps gone rogue: Maintaining personal privacy in an epidemic. *arXiv preprint arXiv:2003.08567*, 2020.

[8] In coronavirus fight, china gives citizens a color code, with red flags. https://www.nytimes.com/2020/03/01/business/china-coronavirus-surveillance.html. Accessed: 05-04-2020.

[9] Alexander Klimburg, Louk Faesen, Paul Verhagen, and Philipp Mirtl. Pandemic mitigation in the digital age digital epidemiological measures to combat the coronavirus pandemic. https://www.aies.at/download/2020/AIES-Studies-2020-12.pdf, 2020.

[10] Greg Walton. *China's golden shield: corporations and the development of surveillance technology in the People's Republic of China*. Rights & Democracy, 2001.

[11] China's hi-tech war on its muslim minority. https://www.theguardian.com/news/2019/apr/11/china-hi-tech-war-on-muslim-minority-xinjiang-uighurs-surveillance-face-recognition. Accessed: 05-04-2020.

[12] Fred H Cate and James X Dempsey. *Bulk Collection: Systematic Government Access to Private-sector Data*. Oxford University Press, 2017.

[13] China launches coronavirus 'close contact detector' app. https://www.bbc.com/news/technology-51439401. Accessed: 05-04-2020.

[14] Coronavirus cases have dropped sharply in south korea. what's the secret to its success? https://www.sciencemag.org/news/2020/03/coronavirus-cases-have-dropped-sharply-south-korea-whats-secret-its-success. Accessed: 05-04-2020.

[15] Coronamap. https://coronamap.site/. Accessed: 05-04-2020.

[16] South korea is reporting intimate details of covid-19 cases: has it helped? https://www.nature.com/articles/d41586-020-00740-y. Accessed: 05-04-2020.

[17] Contact transmission of covid-19 in south korea: Novel investigation techniques for tracing contacts. *Osong Public Health Res Perspect*, 11(1):60–63, 2020.

[18] Trace together. https://www.gov.sg/article/help-speed-up-contact-tracing-with-tracetogether. Accessed: 05-04-2020.

[19] Bay Jason, Kek Joel, Tan Alvin, Sheng Hau Chai, Yongquan Lai, Tan Janice, and Tang Anh Quy. Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders.

[20] Hyunghoon Cho, Daphne Ippolito, and Yun William Yu. Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-offs. *arXiv e-prints*, page arXiv:2003.11511, March 2020.

[21] Pm lee: the covid-19 situation in singapore (3 apr). https://www.gov.sg/article/pm-lee-hsien-loong-on-the-covid-19-situation-in-singapore-3-apr. Accessed: 05-04-2020.

[22] Given low adoption rate of tracetogether, experts suggest merging with safeentry or other app. https://www.todayonline.com/singapore/given-low-adoption-rate-tracetogether-experts-suggest-merging-safeentry-or-other-apps. Accessed: 10-05-2020.

[23] Robert: Robust and privacy-preserving proximity tracing protocol). https://github.com/ROBERT-proximity-tracing/documents. Accessed: 21-04-2020.

[24] Mobile applications to support contact tracing in the eu's fight against covid-19: Common eu toolbox for member states. https://ec.europa.eu/health/sites/health/files/ehealth/docs/covid-19_apps_en.pdf. Accessed: 20-04-2020.

[25] Kirsten Bock, Christian Ricardo Kühne, Rainer Mühlhoff, Měto R Ost, Jörg Pohle, and Rainer Rehak. Data protection impact assessment for the corona app.

[26] Google beacon platform. https://developers.google.com/beacons. Accessed: 07-04-2020.

[27] Getting started with ibeacons. https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf. Accessed: 07-04-2020.

[28] Eddystone beacons. https://developers.google.com/beacons/eddystone. Accessed: 07-04-2020.

[29] How will contact tracing operate in flanders? https://www.vrt.be/vrtnws/en/2020/05/05/how-will-contact-tracing-operate-in-flanders/. Accessed: 07-05-2020.

[30] eidas. https://ec.europa.eu/digital-single-market/en/news/cross-border-digital-identification-eu-countries-major-step-trusted-digital-single-market. Accessed: 07-04-2020.

[31] Serge Vaudenay. Analysis of DP3T (Between Scylla and Charybdis). Technical report.

[32] Sashank Narain, Triet D Vo-Huu, Kenneth Block, and Guevara Noubir. Inferring user routes and locations using zero-permission mobile sensors. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 397–413. IEEE, 2016.

[33] Covid-19 tracking data should be managed the way data trusts are. `https://policyoptions.irpp.org/magazines/april-2020/covid-19-tracking-data-should-be-managed-the-way-data-trusts-are/`. Accessed: 12-05-2020.

[34] Hao Chen, Kim Laine, and Peter Rindal. Fast private set intersection from homomorphic encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 1243–1255, New York, NY, USA, 2017. Association for Computing Machinery.

[35] What we do and don't know about 2 important factors in covid-19's spread. `https://www.businessinsider.com/importance-of-infectious-dose-viral-load-in-spread-of-coronavirus-2020-4?r=US&IR=T`. Accessed: 01-05-2020.

[36] The importance of equity in contact tracing. `https://www.lawfareblog.com/importance-equity-contact-tracing`. Accessed: 12-05-2020.