SafeTrace API

safetraceapi.org

March 24, 2020

1 Introduction

Facing this tremendous global pandemic, it is clear that governmental institutions have fallen very short. On behalf of all concerned professionals and academics globally, our organization is working to fill the gaping technological void in this response. To meet this challenge, we are developing a scaleable API framework, SafeTrace. This decentralized database enables the widespread adoption of contact tracing via mobile devices. With this technology, the world can trace the coronavirus outbreak at the *level of individuals*. Yet, with this compelling technology comes a *serious concern of user privacy*. The key distinction of our endeavor is the level of privacy SafeTrace can provide. Complete user privacy is achieved using state of the art Multi-Party Computation protocols. This whitepaper will proceed by first motivating the necessity of this technology (Section 2), followed by technical background (Section 3), and an overview of our solution (Section 4 and beyond).

2 Motivation

A recent, comprehensive report [1] has been released by epidemiological experts of Imperial College London which is quite sobering. The paper analyzes the estimated effects of various NPIs on the spread of the SARS-CoV-2 virus. Standard non-pharmaceutical interventions include measures such as case isolation, social distancing, school closures, etc. Within this work, two levels of containment are considered and analyzed:

1. *Outbreak mitigation* ($R_0 > 1$):
traditional measures to reduce the rate of infection but not stop it entirely.

2. *Outbreak suppression* ($R_0 < 1$):
more extreme, but still traditional measures to completely stop the spread of disease for as long as possible.

Assuming that a vaccine is not developed in the near future, the effect of *outbreak mitigation* is shown to reduce cases and deaths to 2/3 and 1/2 the amounts estimated without any measures. However, these measures would still result in an estimated *250,000* and *1 million* deaths in the UK and US respectively. See Figure 1 for more details.

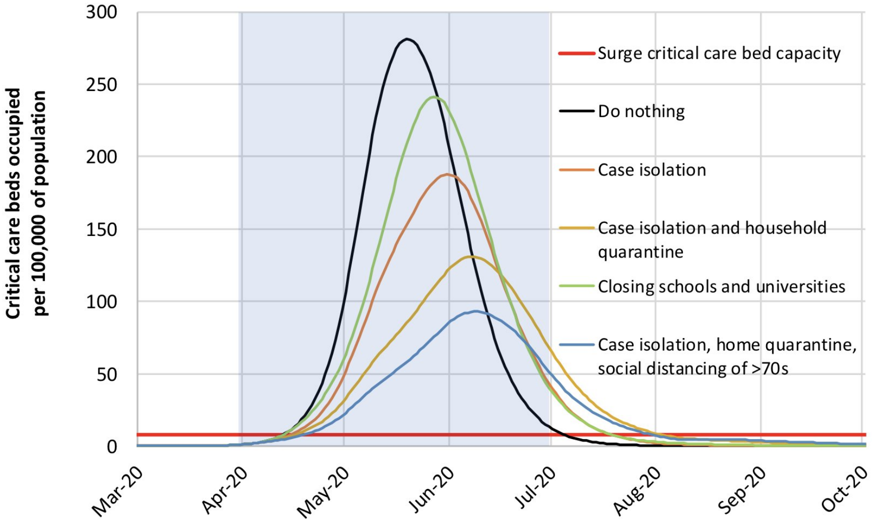

Figure 1: Effects of varying degrees of *outbreak mitigation*. Note that containment measures are present only in the shaded window [1] .

This is unacceptable; therefore, *outbreak suppression* via social distancing is the remaining option. Unfortunately, this work shows that *outbreak suppression* using standard measures is *not a long term solution*. It is clear that suppression cannot be maintained indefinitely due to the economic and social implications. Furthermore, because population wide immunity is not achieved through suppression, this work estimates a rapid surge of cases in the months that follow. See Figure 2 for more details.

The bottom line: Traditional epidemic containment measures are not sufficient to get us out of this mess. In response, we must turn to technological solutions to contain the Covid-19 epidemic.

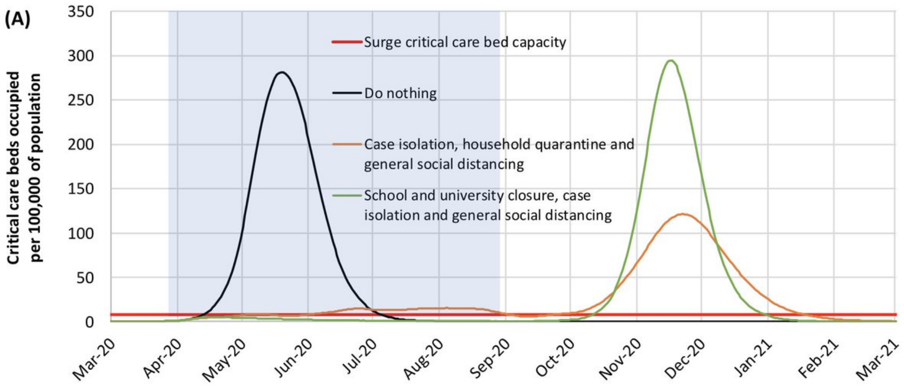

Figure 2: Effects of varying degrees of *outbreak suppression*. Note that containment measures are present only in the shaded window [1] .

3 Background

3.a Contact Tracing via Mobile Devices

As a solution to the ineffectiveness of traditional epidemic containment methods, the academic research community has repeatedly called for the adoption of automated *contact tracing* [3]. Put simply, *contact tracing* is the means by which the effect of an infectious individual is traced through the community. Manual methods are slow and potentially unreliable. In contrast, location data from mobile devices can be used as a medium to accurately track the spread of disease at the level of person to person interactions. This allows individuals to more accurately assess their own personal risk of infection and allow governments to apply interventions intelligently. As soon as someone is tested positive for covid-19, their social interaction network can be notified. See Figure 3 for an example.

3.b Contact Sensing Modalities

To track the spread of a disease, the number of sensing modalities available through mobile devices is quite extensive. The most commonly considered data sources are:

- Bluetooth sensing
- GPS History
- QR code scans

Together, these sensing mediums serve complementary purposes. We will now briefly discuss each in further detail:

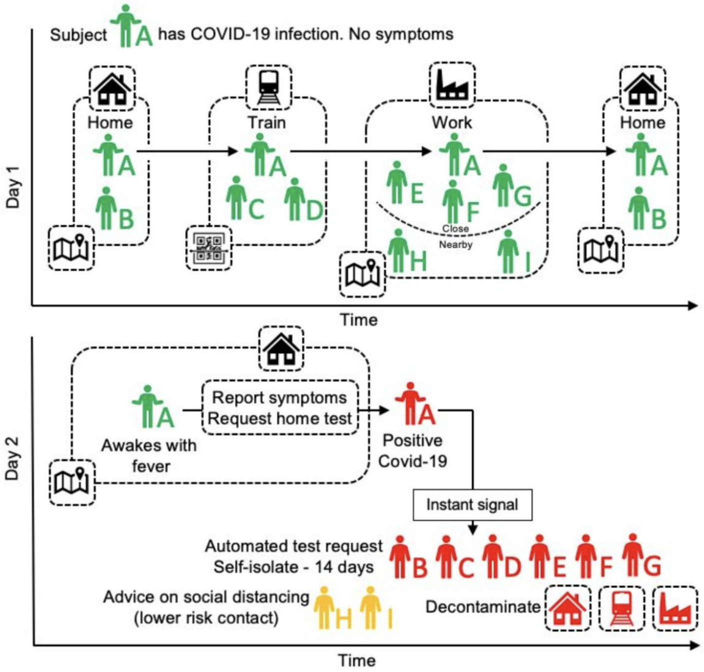

Figure 3: Diagram from [3] demonstrating how a mobile-phone based contact tracing system can quickly notify individuals who are at risk of infection.

3.b.1 Bluetooth Sensing

Bluetooth is used in the context of contact tracing by searching for other users that are broadcasting. In this way, bluetooth is a reactive measure in its role of contact tracing. It keeps track of interactions passively while a user navigates their day to day life. Importantly, it has the capability to measure proximity to other users which is a primary risk factor in contraction of disease. Using bluetooth in this way is akin to standard contact tracing methods; however, it can be done in a manner which is much faster and more reliable. For these reasons, we believe that this is an essential sensing modality.

3.b.2 GPS History

GPS location history is less helpful for tracing interactions at the individual level and more useful for tracking macroscopic trends of infection. This is mainly due to the noise with which GPS coordinates are collected. Similar to the use of QR codes, collecting GPS history data informed by medical prognosis allows users to proactively assess their risk while traveling.

3.b.3 QR Code Scans

This sensing modality has been used to great effect by the Chinese government in their efforts to mitigate covid-19. Essentially, a unique QR code is assigned to each building, taxi, bus, etc. and users scan these codes as they enter and leave. This system is useful as it is robust and provides context to interactions. If two people are in close proximity, it matters if they are inside a small restaurant or out in public. Additionally, this strategy can potentially allow users to assess their risk before traveling which is appealing.

3.c Multi-Party Computation

It must be stressed that with this resource comes the *serious concern of user privacy*. *SafeTrace* is the first scaleable endeavour which holds the potential to provide *complete* privacy for individuals who opt-in to contact tracing services. This privacy is made possible by using SoA multi-party computation (MPC) protocols. We would like to stress, in this context, privacy *completeness* denotes a well defined notion of security. MPC algorithms distribute computation securely over non-colluding entities to provide the same level of security afforded by a perfectly trustworthy third party. To get a sense of where these privacy guarantees come from, we present a brief overview as well as a simple example.

3.c.1 MPC Overview

Multi Party Computation is a set of cryptographic methods for a group of parties to jointly compute a function over private inputs. There are many different MPC constructions but the most classical (and type we are currently implementing) uses Shamir's Secret Sharing as it's primitive, which is a secret sharing scheme based on polynomial interpolation in a finite field. In SSS a value v is split into n shares where any t+1 of these shares put together can reconstruct v, however any t or fewer shares can reveal nothing about this underlying value v. To get a sense for how this is actually possible using polynomials let's walk through an example where t=2: Any 3 (t+1) points will uniquely define a given parabola. However, given only two points on this curve, there are infinitely many possible parabolas passing through them. If four parties each have a different point on a specific but unknown parabola, then any three parties can bring together their points and reconstruct the parabola (from which the secret value can be obtained).

Shamir's Secret Sharing is used for Multi Party Computation because shares are additively and multiplicatively homomorphic (multiplicative homomorphism is a bit more involved but can still be done). If adding and multiplying shares corresponds to adding and multiplying the underlying plaintext secret values, then we can evaluate any circuit as a list of addition and multiplication gates over these ”secret share” inputs. At the end of the computation if parties bring together their result shares, they can decrypt the results of the computation, without leaking any information about the inputs or any intermediate value within the computation.

3.c.2 Motivating MPC Example

To see the applicability of secure function evaluation via MPC protocols, imagine two (not necessarily allied) nations A and B. Both nations have satellites in the atmosphere and it's in the interest of both nations to not reveal the location of their satellites to the intelligence agency of the other. However, it would be a billion dollar loss to both A and B if their satellites happen to collide (this has actually happened). With Multi Party Computation, A and B can jointly execute a computation over mutually held ”shares” of their inputs (the satellite trajectories). The computation can publicly output whether or not the satellites will collide, while keeping the actual trajectory of A or B's satellite wholly private from the opposite party. The nations can collaborate with each other to obtain outputs relevant to both parties (if there is a collision) without revealing any of their personal inputs to each other (where their satellites will be). This is just a taste of the domain of applicability of multi party computation.

4 SafeTrace API

Within this section, we introduce the basic mechanics behind our privatized interface. The macroscopic architecture of our system is similar in spirit to that of Sharemind [2], a well-established protocol in the MPC literature. Armed with these powerful computation protocols, the *SafeTrace* framework will allow for private, decentralized storage of user location data to be used for state of the art contact tracing. This enables individuals, companies, and government institutions to effectively tackle the covid-19 epidemic we are currently facing. To get a sense of how this system will operate, we provide a clear problem abstraction in Section 4.a. Then, we sketch the components of an MPC protocol which solves this problem in Section 4.b. Finally, we dive into the details of how data will be collected, processed, and used to inform users of their risk of infection.

4.a Problem Abstraction

Assume we have n users $U = \{u_1, u_2, ..., u_n\}$. Each user owns associated data $\mathcal{D} = \{d_1, d_2, ..., d_n\}$. Assume further that there exists m servers $S_1, S_2, ..., S_m$, where each server S_i is operated by a distinct non-colluding entity E_i. We will assume that each entity E_i is honest, but curious. This means that each E_i follows the protocol prescribed to it; however, each will try to learn as much as possible about user data in running the protocol. Now, the goal is to design a protocol in which these m entities cooperatively compute some user desired program $P(d_1, d_2, ..., d_n)$, without gaining direct knowledge of user data. Furthermore, we wish the protocol to be *secure*. That is, no user obtains any more knowledge about other users' private data than could be obtained if all computation were done through a trusted third party. Please see Figure 4 for a schematic of this abstraction.

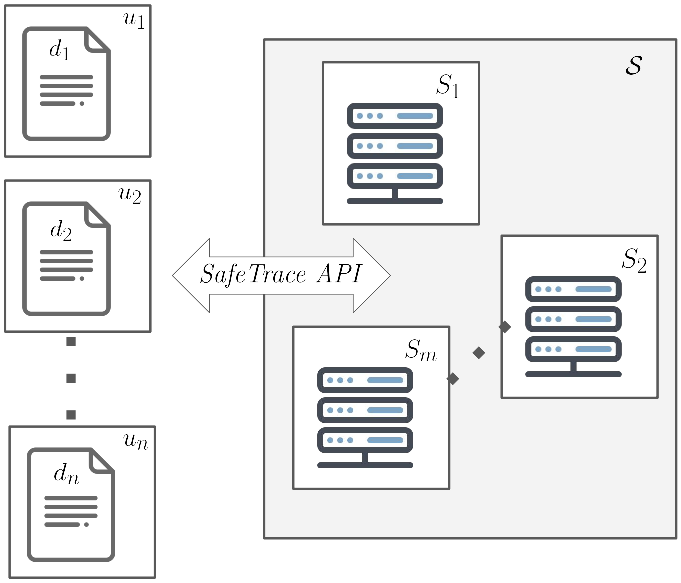

Figure 4: Diagram demonstrating the relationship between users, their private data, *SafeTrace API*, and a set of decentralized storage mediums. Note that the privacy guarantees of *SafeTrace API* allows users to the process their data as though there is one proxy server $\mathcal{S}$ run by a trustworthy third party.

4.b MPC Protocol Sketch

From a high level the MPC protocol will work as follows. End users with personal data can opt-in to a specific computation which will securely aggregate a number of user's data and output the results to these data sharing participants. When a user opts-in they actually Shamir Secret Share each bit of their personal data, and send an array of bit shares to each of the designated MPC computing parties, who will be trustworthy servers all controlled by different entities and on different environments/cloud infrastructure (the non-collusion of these computing parties is paramount to keep data private during computation). The servers collect shares from a number of end users opting into a given computation. To run the computation, the MPC servers all execute the same boolean circuit in parallel, using the collected shares of user data as input wires to the circuit. To evaluate one of these circuits the parties have to do local computations and also rounds of point to point communication and message gathering with one another. At the end of the protocol each of the computing parties has Shamir Secret Shares of the result of the computation, which still don't reveal the results (the parties that compute could be made to never see even the results they computed). The computing parties all send their shares of the results back to each participant. When participants get result shares from a large enough subset of the computing parties, they can locally reconstruct the values of the output bits and finally see the results. In essence, from the perspective of the MPC servers there are these phases:

1. Agree on the parties involved in an MPC operation and agree on the algorithm (circuit) being computed.
2. Collect Shamir Secret Share inputs from end users opting-in to give their data as input to the given circuit.
3. Run MPC protocol with user inputs. This involves iterative rounds of communication and local computations for the MPC servers and results in a set of Shamir Secret Share outputs.
4. Broadcast output shares, or return point to point to specific set of users.
5. Finally end users can reconstruct the results privately and the MPC servers delete all shares of user inputs.

4.c Contact Tracing Data and Risk Assessment

With the computation protocol outlined in Section 4.b, one can now view the collection of servers $S_1, S_2, ..., S_m$ as a single resource for contact tracing. Thus, for this section it is sufficient to consider users having access to a single proxy server $\mathcal{S}$. Now, we would like to illustrate the details of how location data collection, MPC protocol, and risk assessment might come together within a simple model .

Simple Collision Model

- User Data: Let each user u_i possess a unique identifier, U_i. Let user data d_i possesses three fields: $\mathbb{1}_i$, T_i, and a list holding location data. That is, $d_i = \{\mathbb{1}_i, T_i, \{\cdots\}\}$. Let $\mathbb{1}_i \in \{0, 1\}$ be an indicator that u_i has the disease. Let T_i be an estimated timestamp indicating from when u_i was believed to be infectious.
- Database Structure: Let a user's unique identifier U_i and their data d_i form a (key, value) pair for server $\mathcal{S}$. That is, $\mathcal{S}[U_i] = d_i$.
- Bluetooth Sensing: A user u_A is assumed to be able to exactly measure their distance to another user u_B via bluetooth signal.
- Interaction Logging: If this distance $\Delta(u_A, u_B) \leq \delta$ for a time period $\Delta t \geq \tau$ (given appropriate constants δ, τ), then the users log this (δ, τ)-encounter. That is, user u_A appends the following tuple (U_B, t_0) to their location data list while u_B appends (U_A, t_0). Note that t_0 denotes the timestamp when users A and B first crossed the distance threshold δ.

- Risk Assessment: Although it is oversimplistic, let the system suggest that someone be tested if they have ever had a (δ, τ)-encounter. Thus, the desired program $P(\cdot)$ from our abstraction returns 1 if the user has had a (δ, τ)-encounter and 0 otherwise.

Given this model, it becomes clear that a simple notion of risk assessment can be carried out by checking for time dependent collisions in dataset D. Here, even the simplest automated approach nearly matches the level of sophistication of manual contact tracing (with the added speed and accuracy of automation). Now, there is an important question which has been swept under the rug: is program $P(\cdot)$ simple enough to scale well under the MPC protocols? In this instance, it seems likely that the answer is yes; however, if the level of risk assessment is made more complex the answer is no longer as clear.

This is the core technical challenge of our approach, finding a solution which:

1. accurately detects the spread of covid-19
2. scales to millions (or even billions) of users
3. maintains complete user data privacy

5 Key Challenges

Within this section, we would like to convey the hurdles that we will inevitably face in developing and distributing our decentralized framework. With industry backing, we are certain that the following challenges can be surmounted:

1. Providing users ease of access to SafeTrace resources by presenting a simple way to opt-in via established application mediums
2. Gaining public trust in our privacy-first solution via industry backing
3. Organizing the direction of industry and academic leaders to distribute the onus of trust for privacy-first contact tracing
4. Developing the complexity of privacy-first contact tracing while still retaining scalability
5. Acquiring sufficient computation and storage resources to scale SafeTrace to millions of users

6 Epidemic Modeling

As we develop our algorithms further, we plan to incorporate infection models with increasing levels of complexity. Our end goal for modeling will be informed by lead epidemiologists and state of the art research. To get a sense of a more complex model for virus transformation, we present the following framework based on simple SEIR models (Please see *"Proposed Epidemiological Models and Methods to Improve COVID-19 Contact Tracing and Outbreak Prediction"* for more in depth details on infection modeling).

6.a Building a Network of Individuals

Each user connected to SafeTrace can be viewed as a single node within a social network. Every time two nodes come into contact (within some distance threshold), an edge will be created between them. The assumption will be made that every pair of nodes is able to infect or receive infection with each connected node, meaning the graph will be a symmetric directed graph. At the initial time (t=0) each node will be categorized as susceptible, exposed, infectious, or recovered based on current user information. At each subsequent time point, nodes and edges will be changed based on interaction and user input to represent movement of individuals and infection. The result of this process being run at each pre-determined time point forms a dynamic graph that represents the movement of exposure, disease, and recovery within groups of individuals over time. An epidemiologic model—in this case an SEIR model—can be applied over the network to evaluate and predict flow of disease.

6.b Modeling Disease Spread via SEIR Models

An SEIR (susceptible, exposed, infectious, recovered) model will allow tracking of the number of individuals in each of these categories. Flow between each of these categories are modeled through a system of differential equations. See additional, internal whitepaper on details for the design choices and equations behind this model. This basic SEIR model is a "mean-field model,‘' which means that all members of the population receive the same mean treatment. Therefore, the baseline model struggles to capture the complex and heterogenous structures of real-life networks and interaction.

As more data are gathered on individuals and their behavior, it is obviously desirable to take such heterogeneity into account. A crude approach to handling this would involve partitioning the network of all individuals into subnets (by location or by strata of infectiousness), evaluating infection dynamics within each subnet at each time step, and then move the relevant nodes, with their anticipated state at the next time step, to different subnets. This crude process limits the analysis that can be done and makes it more difficult to perform population-level analysis. A more elegant solution—involving pairwise approximation methods applied to the SEIR model—has been explored in recent years. Using this method, each pair of nodes is evaluated at each time step, and therefore differential dynamics between connected nodes can be effectively analyzed. Each node has its own parameters, such as infectiousness, that are incorporated into each pair of nodes' interactions.

A priori understanding of some of the SEIR model parameters and context-specific infectiousness, among other values, help inform the model design and implementation. This is an important initial step now that the model has been laid out, especially since initial use of the API, and therefore data input, will likely be low at initial release. In addition, some parameters such as level of infectiousness or how long the virus remains at a given location can be potentially learned through the continuous use of machine learning algorithms on the data that is gathered.

7 Professional Affiliations

During our development of SafeTrace, we have relied on the mentorship and backing of experts in epidemiology, as well as on research contacts in the MPC community. We are primarily backed by the University of Cambridge in our endeavours. We would like to extend our thanks to Dr. Yoneki and Dr. Crowcroft at the University of Cambridge for their expertise and continued advisory work. In 2010, the research duo launched The FluPhone Project, the first national initiative to employ mobile contact tracing methods. We have assumed responsibility over FluPhone's original codebase, user documents, technical specs, and more. SafeTrace is the next natural iteration on this idea. With our global-facing solution, the academic community can appropriately extract data that is critical to understanding this outbreak as it unfolds.

References

[1] Neil M Ferguson et. al. "Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand". In: *MRC Centre for Global Infectious Disease Analysis* (2020). URL: https://www.imperial.ac.uk/mrc-global-infectious-disease-analysis/news--wuhan-coronavirus/.

[2] Willemson J. Bogdanov D. Laur S. "Sharemind: A Framework for Fast Privacy-Preserving Computations". In: *Computer Security - ESORICS* 5283 (2008). URL: https://link.springer.com/chapter/10.1007/978-3-540-88313-5_13.

[3] Fraser Bonsall Parker. *Sustainable containment of COVID-19 using smartphones in China: Scientific and ethical underpinnings for implementation of similar approaches in other settings*. Mar. 2020. URL: https://github.com/BDI-pathogens/covid-19_instant_tracing.