Community-driven contact tracing using TraceTogether

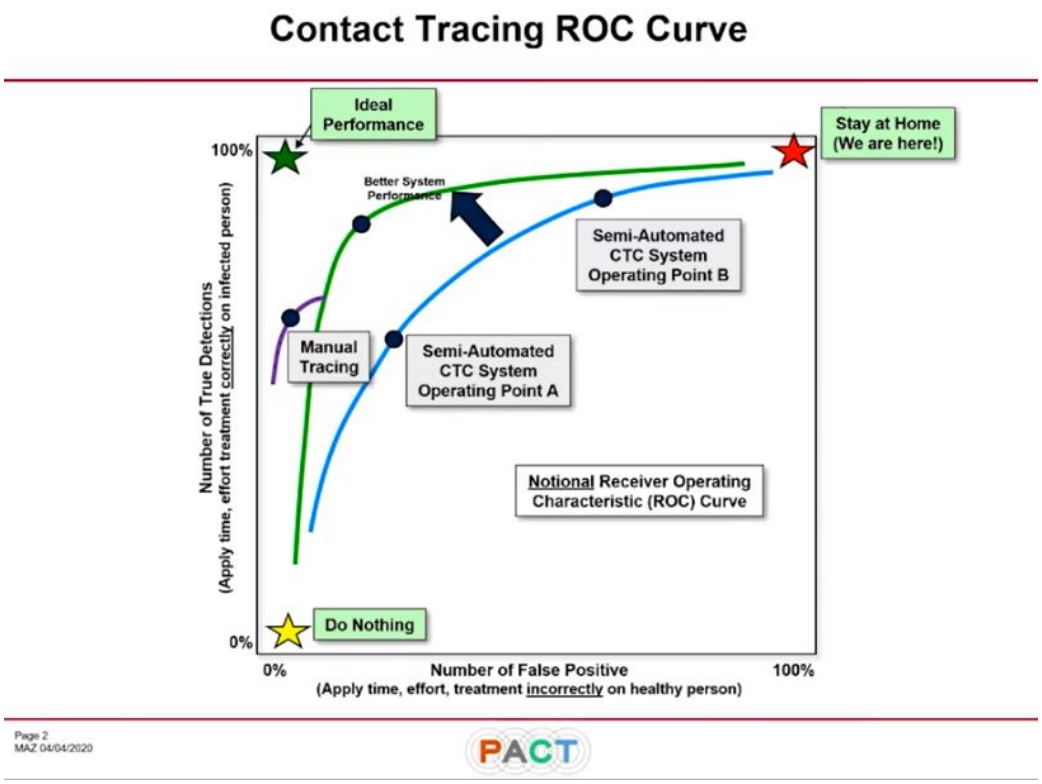
This is part of a joint work with Maha Ashour, this is the part I wrote.

Introduction

Contact tracing is the tracking down of people that were exposed to an individual infected with a virus while they were contagious [1]. It is an important public health intervention that has taken on a renewed urgency in light of the 2019-20 coronavirus disease (COVID-19) pandemic [2]. The aim is to lower the burden on healthcare systems due to the virus while incurring smaller economic costs than that of the population wide self quarantine measures being implemented. Contact tracing does this by enabling targeted quarantines and faster testing and isolation of infected individuals.

Automatic contact tracing is meant to be a useful and cost-effective supplement to manual contact tracing. The latter is limited by imperfect human recall and/or incomplete information due to the difficulty in identifying unknown contacts or 'strangers'. Automatic contract tracing can be used to handle a large percentage of the more simple cases so that the limited resources for manual tracing can be directed towards the minority of more complex cases. The primary goal of privacy preserving contact tracing is to facilitate these efforts without putting into place - and normalising - deeper governmental surveillance on civilians. What exactly I mean by privacy preserving will be discussed in the next section.

The picture below is from the slides of John D Wilkinson presented at a recent MIT webinar on contact tracing [3]. It shows a receiver operating characteristic (ROC) curve. An ROC curve for a diagnostic system is a plot of true positive rate against false positive rate. The area under the curve illustrates the efficacy of the diagnostic system. In the context of contact tracing our goal is to create a diagnostic system for the people infected by individuals that recently tested positive. Here we want the diagnostic system where manual contact tracing is aided by automatic contact tracing to be a better diagnostic system than that made possible by manual tracing alone. I will therefore use the ROC curve to define the correctness of automatic contact tracing.



Contact tracing is much easier and more efficient when we have a shelter in place order in effect. It is much harder to manually remember, list, and track down all the people that may have been infected by an individual when everyone is going about their daily life as normal. It is important to note therefore, that when we talk about automatic contact tracing we're talking about a technology that can be used with minimal disruption to everyday life.

A big factor in the efficacy of contact tracing is adoption rate. Although this is not a security concern per say, it is important to keep ease of adoption and incentives against adoption in mind when developing such a protocol.

Security Definition

The participants of the protocol are a group of central trusted authorities and then some number of users (i.e the citizens of the country/countries where this measure is being implemented). The central trusted authorities have no input but each user does have an input (phone number, location, etc.).

Contact tracing produces a list of individuals as its output. These individuals are considered those that most likely contracted the virus from an infected individual (someone who recently tested positive) and will count ultimately as either true positives or false positives on the ROC curve . The output of contact tracing aims to aid manual tracing efforts so that we end up with a

better ROC curve. This can mean several things. For one, automatic contact tracing can potentially catch cases missed by manual tracing (individuals unknown to or unnoticed by the person who tested positive, in addition to people they might have forgotten). Automatic contact tracing can also be a resource efficient way to catch a large percentage of the easy cases leaving manual tracing with more resources to catch the more involved or complex cases.

The security considerations of the protocol can be broken up into two types, namely privacy and correctness, each of which is addressed below.

Correctness

The list output by the protocol is meant to supplement interviews and other efforts of manual contact tracing but not meant to bypass them by any means. Therefore, the measure of correctness is that manual contact tracing supplemented by the output of the protocol does not land us in a worse part of the ROC curve than manual tracing alone. Any particular solution aims to diagnose more true positives without increasing the ratio of false positives.

Privacy

Privacy preservation for contact tracing refers to the protection of the users' inputs. We will break this up into two components. Protection from the central 'trusted' authority, and protection from other users.

Protection from central authority:

Contact tracing prevents the central 'trusted' authorities from learning information about the users' inputs other than what is revealed by the output of the protocol. This includes being able to track the location of individuals outside of any location information revealed by the output of the protocol. This definition also implies that the central authorities should not learn the phone numbers of at least the individuals that they do not need to contact. TraceTogether's current implementation reveals the phone numbers of all users to the central authorities. We note however, that the whitepaper states the protocol can be implemented without using phone numbers at all. Depending on how this is done, it could satisfy privacy from the central authority as defined here. If not, a future analysis may want to include that the output of the protocol to the central authorities includes the contact information of all participating users.

In particular, contact tracing prevents the central authority(ies) from collecting location information on users. It prevents the collection of personally identifiable information, or tracking of the proximity of pairs or groups of users over time, etc. except for pieces of information that may be revealed by the output list of individuals. Importantly, privacy preserving automatic contact tracing does not prevent any trusted central authority from affecting correctness, indeed the central 'trusted' authorities are trusted to not tamper with the protocol in such a way (i.e by causing the created lists to be longer or shorter than they would have been in the absence of tampering).

Protection from other users:

Automatic contact tracing aims to prevent users from learning information on other users' inputs outside of the information that can be learned from the output of the protocol. It is not clear whether in any particular protocol, users will even learn the output of the protocol, but for the sake of this paper let's say that they learn at most the output of the protocol. This is because users will learn about the people they know, who have been asked by the government to self quarantine and get tested, and this will likely reveal some information about the output of the protocol in some cases.

Note for analysis: This seems like it would include phone models but we can say that users can learn the phone models of other users simply by looking when they were in contact with them, so technically this need not be additional information revealed by the protocol. Although in some cases it is but then indistinguishability should maybe still work out if we assume people can't learn model numbers without the protocol in only a negligible number of cases?

There isn't a standard problem definition for this problem as the research on this topic is evolving everyday, even as I write this paper. Our problem definition agrees with the problem definitions from other papers that are currently available on the topic. For example, [4] define the problem as having the goal of data minimization. That is, the central authority should learn the minimum possible information about the users of the app. We assert that this goal is satisfied by a solution that does not reveal any information on the users' inputs to the central authorities outside of the list of infected individuals.

TraceTogether: one particular protocol

TraceTogether is the Singapore government's proposed solution to the problem of private automatic contact tracing. It solves the problem of imperfect or incomplete human recall in manual contact tracing by using Bluetooth technology in mobile devices to track a user's close contacts. The app works by identifying who a user may have in close contact with, as opposed to where the user has been (for example, using location history). The privacy- preserving protocol that enables this contact tracing is called BlueTrace.

The use of mobile phones and Bluetooth in TraceTogether is an important design choice. The ubiquity of both technologies allows for quick and widespread adoption of the program, as well as a high degree of interoperability between devices. Moreover the reliance on Bluetooth instead of location history provides more fine-grained and precise distance information than would have been possible using cell-tower and GPS data (for instance, correctly not co-locating two people in the same building on different floors). Furthermore, decisions about how to achieve certain security goals in the protocol depend on the efficiency of the protocol for feasibility and time

considerations but also for battery drainage reasons. Bluetooth succeeds in this regard as it consumes a very small amount of power, in particular it consumes less power than the other most likely candidate: wifi. The goal is for users to have as little incentive as possible to opt out of using the protocol as possible.

TraceTogether app users register with the government using their phone number. When users are in close contact with others they exchange randomized tokens. These tokens together form the user's encounter history and are stored locally on the phone. The goal of these tokens is to hide the user's personally identifiable information (PII) and location information from other users and even the central trusted authority. If a user later tests positive for the disease, the health authorities will contact that user and request that the user upload their encounter history to the government's central servers. The government maintains a database which allows them to link the tokens to individuals and their phone numbers (So while the different app users can maintain anonymity in their interactions, they are not anonymous to the government) [5]. This design also requires a trust system to verify that the request is truly originating from the government server, and as a result a PIN authentication mechanism is used. The details of this are not elaborated in the BlueTrace whitepaper [12], nor is this implemented in the open source code as each government is expected to implement this piece on their own. As a result, we do not have access to how exactly the TraceTogether's implementation handles this and cannot comment on it.

Once the data is received by the central government server, it links them to the phone numbers that were used during the app's registration. At this point, the phone numbers of the user and that of the people they potentially infected are revealed to the government so that it can notify them of possible infection. The TraceTogether approach relies on the assumption that phone numbers do not reveal 'too much' PII. The protocol reveals phone numbers of all users to the central authority. It also reveals the phone numbers of all users potentially infected by a user to the authorities to allow the central authorities to contact these individuals. I repeat, again, that the white paper states that this entire process can be done without phone numbers revealed to the authorities in either phase.

Protocol Details

In this section and the next I will describe the TraceTogether protocol and discuss how this protocol satisfies our problem definition. I use the word discuss here instead of prove on purpose. Given that this protocol is brand new, there does not exist a formal analysis of this protocol under any problem definition, I try to get closer to that ultimate goal here.

The protocol can be seen as divided into four distinct phases for each user:

1. **Initialization**: This is the phase where a user opts into the protocol. Here, the user sends their phone number to the central authorities who then generate a random user id for them and initialize the information required for the creation of tokens phase.
2. **Creation of Tokens:** The token that a particular user sends to other users changes periodically. (This is done to prevent tracking of users and also to prevent malicious users from causing false positives by echoing other users.) The randomized tokens created by the central authority therefore contain information about the validity time period of the tokens in a verifiable and encrypted manner. In this phase the central authority uses the initialized information on the user stored in its database to create a portion of the randomized tokens that the user will exchange with other users it comes into contact with. These tokens are then sent to the user along with the information about when each token will be valid. The authorities send a list of tokens to the user for future times to ensure that the protocol can continue to work even when users are not connected to the authority (for example if they go outside the range of their WiFi connection). This phase repeats when the user runs out of tokens. It is not clear from the whitepaper or the code how the API calls requesting new tokens are screened at the server's end. If the server does not prevent the user from continually spamming it by asking for new tokens, this may cause blocks in the code, but it also may cause issues related to key reuse etc.
3. **Exchange of tokens:** Whenever a user is near another user, they exchange tokens via bluetooth. Each user sends its currently valid token to the other. The tokens that a user receives along with the time the user received the tokens are stored locally at the user's end. Identifiers are exchanged between nearby phones using Bluetooth which is a low-power signal that quickly degrades over distance. Therefore the signal strength can be used to estimate the distance between two devices. In order for the app to work, Bluetooth advertising has to be enabled on all app users' phones.
4. **Infection:** When a user is infected, the central authorities request the user to upload their local database of received tokens from the time period when they were contagious. To prevent users from uploading this information without being tested positive, the whitepaper suggests that the request for uploading tokens will contain some PIN that the user can use to upload the tokens. As discussed earlier, the method for doing this is neither discussed in the whitepaper nor is it implemented in the available code, therefore, we cannot comment on its efficacy but rather we assume that it satisfies its stated purpose. Once the authorities receive the tokens uploaded by a use they decode and remove any tokens that were received when they were not valid. They then can see which phone number belongs to and use this information to determine if any users were close enough to the infected user for long enough to be infected themselves. At this point the white paper uses a human in the loop approach to check this list against the interview of the infected user for anything the protocol might have missed. The human can classify which users may have been infected by checking the interview against the decoded information better than an algorithm by catching nuances that a quickly thrown together algorithm may not. Furthermore a human can check the decoded information against the interview to prevent some obvious malicious attacks for example one where the infected user uploads tokens received by him but also tokens received by a whole

coalition of malicious users. A human could immediately recognize that this list is much longer than it should be given the places the user said they had been in their interview.

Details left out from the above description:

Tokens: Each token consists of a 21 byte temporary id and start and expiration time stamps, which are together symmetrically encrypted using AED-256-GCM. Galois Counter Mode (GCM) is a mode of operation for symmetric-key encryption that is used with a 128-bit block cipher in counter mode; AES in this case. The result of this authenticated encryption scheme is a ciphertext as well as an authentication tag which is used to ensure integrity. The encrypted cipher text is concatenated with the random initialization vector (IV) and the authentication tag, and the entire message is then base64 encoded. The resulting string represents a single identifier. Using GCM to encrypt the identifiers ensures protection of data at rest against an active adversary as specified by the security definition of authenticated encryption. Since only the government holds the encryption key, the privacy of the app users is protected from other users.

The ephemeral identifiers also help to prevent a replay attack, where a user receives identifiers and then re-transmits those identifiers to other users at a later time. Since both the expiration time and identifier are encrypted, a re-transmitted message can be quickly detected, and the window for such an attack is severely limited to the 15 minutes during which the ephemeral id is valid. Moreover, the time that two users have to be near each other to cause an infection is greater than 15 minutes so a malicious user would have to replay consecutive tokens to cause a false positive. The authentication tag also prevents a malicious user from crafting new identifiers.
If a user is isolated and has contact with only one other person a linkage attack is possible if the user is later notified that a close contact is infected; there is only one such possible contact and therefore anonymity cannot be maintained. However, since such an attack by definition reveals no more information than what is revealed by the output of the protocol, it satisfies our problem definition.

Classifying close contact: Close contact is defined as being within 2 meters for 30 minutes or more. The actual value is chosen to balance between true positive and false positive rates. There is a trade-off between accuracy on the one hand and power and privacy on the other hand. In order for the Bluetooth signal to be used to estimate distance, multiple measurements are needed because of variations in the signal strength resulting from the orientation of the phone at the moment it was receiving the signal or it's placement in a pocket or relative to the body. However Bluetooth scanning is more power intensive and this has a negative impact on battery life. Moreover the repeated measurements may enable tracking of users. Furthermore, the human in the loop method is used to identify the cases where infection occurred due to close contact and this can make up for some issues with this definition of close contact.

It is worth noting here that automatic contact tracing is most beneficial when a user is carrying their phone with them at all times. However based on our security definition if users do not

always behave in this way it will not lead to less correctness than relying on manual tracing alone. The same could be said for an attack that jams bluetooth signals or a user who deletes/tampers with their locally stored data. That may cause missing/incomplete information but won't reduce the "correctness" over what manual tracing can provide.

Some other notes:

One attack that most solutions to automatic privacy preserving contact tracing are vulnerable to is the Sybil attack. This attack involves the creation of multiple accounts on multiple devices by a user. I would like to note that the use of phone numbers in this solution to register app users, while possibly problematic for other reasons, could help mitigate such an attack.

Two users can still cause a false positive by 'targeting' someone. One user remains in proximity to the target, echoing the target's tokens to another user, who can then store them and or echo them to the users around him. If any of these users then test positive, the target will be identified as being infected. One needs to be careful when checking that this still satisfies correctness.

The whitepaper for the protocol does not state whether the time periods for the validity of the tokens are synced across users or different for each user. Either choice may cause minor issues but I do not go into the details of this due to lack of space and time.

[1] What is contact tracing?
https://www.who.int/news-room/q-a-detail/contact-tracing

[2] Whats the coronavirus disease pandemic?
https://www.who.int/emergencies/diseases/novel-coronavirus-2019

[3] John D. Wilkinson, MIT Lincoln Laboratory, ImPACT 2020 webinar
https://web.mit.edu/webcast/pact/s20/

[4] The EU solution: DP-3T
https://github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf

[5] Paper analyzing TraceTogehter
https://arxiv.org/pdf/2003.11511v2.pdf

[6] NIST recommendations for GCM (section 8.3)
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

[7] Issue page on github with key invocation problem was raised
https://github.com/opentrace-community/opentrace-cloud-functions/issues/7

[8] CVE-2020-11872

https://nvd.nist.gov/vuln/detail/CVE-2020-11872

[9] Synchronous use of crypto.randomBytes()
https://nodejs.org/uk/docs/guides/dont-block-the-event-loop/#blocking-the-event-loop-node-js-core-modules

[10] Issue page on github where we shared concern about use of synchronous PRNG
https://github.com/opentrace-community/opentrace-cloud-functions/issues/33

[11] Bluetooth advertising implementation
https://github.com/opentrace-community/opentrace-android/blob/master/app/src/main/java/io/bluetrace/opentrace/bluetooth/BLEAdvertiser.kt

[12] BlueTrace white paper
https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf

[13] OpenTrace source code
https://github.com/opentrace-community