Defending Against Linkage Attacks in Decentralized Contact Tracing Protocols

Robert Coleman †

April 20, 2020

Abstract

Large scale decentralized contact tracing protocols with user privacy protections have been proposed by several researchers in an attempt to curb the transmission of COVID-19 and 'flatten the curve'. We surveyed six (6) protocols to understand common design trends and trade-offs, with a focus on their defense against linkage attacks, and without assessing the underlying cryptography or effectiveness in addressing the pandemic.

1 Introduction

As the Coronavirus Disease (COVID-19) has grown into a global pandemic, governments have been forced to shut down businesses and issue stay-at-home orders to citizens. Many researchers are now proposing the use of contact tracing to help reduce the demand for medical services and proactively warn citizens about exposures to COVID-19 before they can spread the virus. To enhance the ability and accuracy of contact tracing, researchers have investigated the incorporation of mobile technology.

1.1 Contact Tracing

Contact tracing is the process of identifying and alerting people that have come in contact with, and potentially contracted, a disease in order to prevent its further spread. In practice contact tracing heavily relies on the collection of proximity data; information on who one has been in close contact with. Traditionally this is done by interviewing patients after a positive test result. However, the process is slow, unreliable, and unscalable. In the context of the protocols that we surveyed, proximity data is collected by mobile devices broadcasting and receiving Bluetooth identifiers. Proximity data is compared to information provided by known disease carriers to produce alerts for people that have been in close contact with contagious persons. This analysis can be conducted in a centralized or decentralized manner, typically by a government agency or on user devices. In the protocols surveyed, a centralized server is used to publish records to user devices where contact tracing is performed in a decentralized manner.

1.2 Linkage Attacks

Linkage attacks are attempts to deanonymize anonymous datasets by correlating them with identified datasets. While the motive for these attacks differ, the result leaves the victims vulnerable to abuse. In the context of COVID-19, targets can face social ridicule, threats of violence, or fall victim to other hostile behaviors.

†robert@robertjcoleman.com

In this survey, six (6) decentralized contact tracing protocols were reviewed to understand their designs, trade-offs, and resultant impacts on the defense against linkage attacks. The survey critically evaluates the protocols' ability to protect users against attacks from small-scale adversaries (e.g. coworkers and neighbors), large-scale adversaries (e.g. governments and large corporations), and network eavesdroppers, and prevent individual records from being linked to one another or specific healthcare providers.

2 Apple/Google Protocol

To promote interoperability between user devices, Apple and Google have partnered together to create a decentralized contact tracing protocol, including a Bluetooth Specification, Cryptography Specification, and Framework API. In this survey, we reviewed the Cryptography Specification [1], where the companies outline how their protocol will handle proximity data collection and conduct contact tracing.

2.1 Overview

The proposed protocol relies on three types of data: a 32-byte Tracing Key, 16-byte Daily Tracing Keys, and 16-byte Rolling Proximity Identifiers. When contact tracing is enabled on the user device, a Tracing Key is generated using a cryptographic random number generator (CRNG). This key is used to generate the Daily Tracing Keys and should never leave the device. Using Hashed Message Authentication Code (HMAC) based key derivation function (HKDF) with SHA-256 hash function, the protocol derives a Daily Tracing Key using the Tracing Key and a 32-bit DayNumber (a number associated with each 24-hour day since the Unix epoch). This method allows for Daily Tracing Keys to be recalculated when needed. From the Daily Tracing Key, the protocol derives a series of Rolling Proximity Identifiers by truncating an HMAC of the 8-bit TimeNumberInterval (a number associated with every ten-minute window in a day) signed with the Daily Tracing Key.

The protocol enables proximity data collection by having users broadcast their Rolling Proximity Identifier in Bluetooth Advertisements while keeping logs of identifiers the user's device has encountered. The Rolling Proximity Identifiers are rotated every 10 minutes to preserve user privacy and prevent tracking.

When a user tests positive for COVID-19, their Daily Tracing Keys and associated DayNumbers are uploaded to a Diagnosis Server. The uploaded data is limited to only the days the infected user was suspected to be contagious. The Diagnosis Server is the centralized server responsible for aggregating and distributing keys and DayNumbers to user devices. With the Daily Tracing Keys and DayNumbers, user devices generate the sequences of Rolling Proximity Identifiers broadcasted by the infected users, and compare them to the identifiers they have encountered.

2.2 Evaluation

To reduce the bandwidth and space constraints associated with scaling this system, the protocol uses linkable Bluetooth identifiers at the cost of infected users' security. If an adversary has information from a network of Bluetooth receivers with known locations, the information provided by the Diagnosis Server can be linked to the Bluetooth network data to identify the movements of an infected user throughout a day. With an area network of Bluetooth receivers required for this attack, this is presumably only feasible by large-scale adversaries. Despite the requirements, the potential for this attack to be successful is quite high, with Bluetooth receiver networks currently existing in cities for several reasons, such as traffic monitoring. Depending on the placement of the receivers and the other

information available from the area (e.g. security footage), this attack could result in the successful identification of infected individuals.

On a small scale, adversaries could easily keep timestamped logs with whom they have been in close contact. Linking this data to observed Bluetooth identifiers would allow an adversary to identify broadcasts by a specific person. Data from the Diagnosis Server would then allow the adversaries to track whether a specific person has tested positive. There is a potential for false negatives with this attack as it requires the target to willingly upload their Daily Tracing Keys (records) to the Diagnosis Server. While the attack also depends on being in proximity to an infected user, a tech-savvy adversary could use a Bluetooth antenna to collect identifiers from a distance.

A protocol for uploading and publishing records to the Diagnosis Server is not included in the specification and therefore the aspects relating to the server and its defenses cannot be properly evaluated.

3 Canetti et al. Protocol

The protocol by Canetti et al. [2], hereinafter referred to as the Canetti et al. Protocol, proposes a different system that addresses some issues regarding linkage attacks which is less formally defined compared to the Apple/Google Protocol.

3.1 Overview

The Canetti et al. Protocol proposes a system where user devices broadcast transient Bluetooth identifiers, or tokens, for a period referred to as a *tick*, which is similar to the Apple/Google Protocol. The protocol suggests tokens be generated from a Pseudo-Random Number Generator (PRNG), a Pseudo-Random Function (PRF), such as AES, or a cryptographic hash function such as SHA-256, with different seed values to ensure that tokens cannot be linked to one another. As a result of this design choice, users must store every token they broadcast over a given retention time, for which the protocol suggests a two weeks.

When a user tests positive for COVID-19, the infected user works with medical staff to upload their recorded tokens to a registry. Since tokens cannot be linked to one another or derived from a common seed value and function, every token broadcasted by an infected user must be uploaded and distributed to all other users, vastly increasing the space complexity, and therefore the bandwidth of the system. To avoid this issue, the Canetti et al. Protocol suggests that either the infected user or medical professionals generate a Bloom Filter with the broadcasted identifiers and upload the resultant filter to the registry where it can incorporate the resultant filter and redistribute the new filter to all devices. A Bloom Filter is a probabilistic data structure used to test set membership. By design, Bloom Filters can only assert that values do not exist in a set or that they may exist in the set with a false positive probability. This is due to their use of hash functions and their potential for collisions.

3.2 Evaluation

The Canetti et al. Protocol design choices better defend infected users from linkage attacks compared to the Apple/Google Protocol. While not impossible, large-scale adversaries cannot easily determine the movements of the target throughout a Bluetooth receiver network due to the unlinkable design of the Bluetooth identifiers in this protocol. Given a dense enough network and sparse enough population, or with reliable broadcast signal strength information, the adversary could infer that an identifier at $tick_n$ and $tick_{n+1}$ belong to the same user if they are in the same relative location at the end and the start of the respective

periods. This methodology is just one of the ways adversaries currently circumvent Bluetooth MAC address randomization to track mobile users. Nonetheless, it is important to note that unlike the Apple/Google Protocol, this protocol does not provide the same degree of certainty and affords infected users more control.

Similar to the Apple/Google Protocol, the Canetti et al. Protocol is not able to defend against small-scale adversaries abusing the system to learn the infection status of users. The protocol does reassure infected users that while such attacks are inevitable, they are hard to scale. In an extension, the Canetti et al. Protocol proposes mitigating this attack by creating a private registry, thus transitioning away from a decentralized system, and falling outside the scope of this survey.

Without the use of Bloom Filters, the system is more vulnerable to adversaries linking together individual records from the registry. Using snapshots of the registry data and the number of records infected user uploads, n, an adversary could determine that the most recent k records belong to 1 of k/n. If k = n, then it becomes trivial to link the records together. Through the proposed use of Bloom Filters, this attack would become computationally infeasible.

To avoid network-level identity leaks, the protocol recommends the use of network-level anonymization systems (such as Tor) be used to upload records to the registry. This will prevent network providers from tracking where new registry records originated from, preventing them from linking identities to infected users, or uploads to specific health-care providers.

4 East Coast PACT Protocol

R. Rivest et al. [3] have published a PACT Protocol which is referred to as the East Coast PACT Protocol in this document. To avoid confusion the PACT Protocol published by J. Chan et al. [4] is referred to as the West Coast PACT Protocol in this document.

4.1 Overview

The East Coast PACT Protocol is divided into two layers (a chirping layer and a tracing layer) which mirror the proximity data collection and contact tracing discussed in the introduction.

A chirp is the name given to a Bluetooth identifier broadcasted over a small subset of time. In the chirping layer users generate a 32-byte seed used to derive a series of chirps over an hour. A 28-byte chirp is calculated using a PRF with the seed and the current time. The resulting chirp is then broadcasted for a defined number of minutes before being rotated to a new value. Synchronously, user devices observe and log chirps in proximity with their broadcast strength and observed time, optionally including the users' current GPS position. The protocol outlines that users should keep logs of all seeds and observed chirps for a medically relevant time period, with three months being suggested by the protocol.

The tracing layer involves the mechanisms for uploading seeds to a central server and performing contact tracing on user devices. The protocol suggests medical professionals be provided with permission numbers that will be dispersed to infected users who have tested positive for COVID-19. These permission numbers will authorize the infected user to upload their seeds to the central server. The infected user will input the permission number to their application and initiate the upload process. Before the data leaves their device, infected users will have the opportunity to censor any of their seeds by replacing them with newly generated seeds for identity protection. The seeds uploaded by the infected users will be accompanied by the time they were used, represented by a start and end time, to reduce the

computational complexity of contact tracing. Similar to the Apple/Google Protocol, users will query the server for seeds that they will then use to generate the chirps broadcasted to compare to their logs. Unlike the other protocols, the use of stored location data and broadcast strength is used to generate a stronger exposure score, where infected users can alter certain locations (such as public transit) with higher scoring weights.

4.2 Evaluation

The East Coast PACT Protocol employs the use of identifiers linkable through a common seed that is distributed to every user similar to the Apple/Google Protocol. As previously discussed, this allows for infected users to be tracked by large-scale adversaries capable of operating a network of Bluetooth receivers with known locations. As such, infected users' commutes or other travel can be effectively and autonomously tracked. In this protocol, the linkability of identifiers is limited to one hour, after which a new seed is used.

Similar to other protocols, this protocol leaves infected users susceptible to being identified by collocated adversaries. As previously mentioned, adversaries can limit their exposure to a single individual to later identify the infected user through the use of the central database. Unlike other protocols, the East Coast PACT Protocol combats this by allowing the infected user to censor their seeds before uploading them, enabling conscientious infected users to protect themselves.

In the outline of the upload mechanism, the protocol attempts to defend against the linking of uploads to individual medical professionals by stating that the permission numbers would be randomly assigned. It is uncertain if this will adequately defend against linkage attacks without the implementation details of this distribution system. The protocol does go on to suggest the use of network-level anonymization systems (such as Tor) to prevent network administrators and providers from identifying infected users by linking upload traffic and IP addresses.

Finally, the protocol does not mention the mechanism used to publish records. As discussed in the Canetti et al. Protocol, this potentially leaves records vulnerable to attack, where the publishing of records in batches allows individual records to be linked together, circumventing the security of rotating underlying seeds.

5 West Coast PACT Protocol

The West Coast PACT Protocol proposes multiple methods for dealing with contact tracing, of which this survey looks at the privacy-sensitive mobile tracing.

5.1 Overview

In the West Coast PACT Protocol, Bluetooth identifiers take the form of 32-byte IDs rotated every time increment. The protocol generates the linkable IDs using a PRF, such as SHA-256, and a 32-byte seed. As in other protocols, seeds are rotated on a fixed schedule to reduce the linkability of infected users. User devices simultaneously broadcast the identifier for the current time increment while listening for and storing IDs and the timestamps of when they were observed.

After testing positive, the infected user will upload a record of their seeds with their corresponding generation time and termination time. It is suggested that these entries be validated with the use of signatures and an optional certificate to validate the signature. Signatures would likely need to be provided by third-party health-care providers to be effective in guaranteeing authenticity. The protocol also suggests that the publishing of records to the database be delayed to prevent rebroadcast attacks.

5.2 Evaluation

The West Coast PACT Protocol does not protect against large-scale adversaries from tracking the short term movements of infected users, similar to the East Coast PACT and the Apple/Google Protocols. Given that broadcasted identifiers are linkable, the movements of a reported infected user can be traced together with a large enough network of Bluetooth receivers.

The protocol does directly address security concerns relating to linkage attacks from small-scale adversaries attempting to link user identities to infection statuses. To mitigate this, the protocol proposes the use of Decisional Diffie-Hellman but immediately notes that an adversary can circumvent this through the use of location-specific seeds.

Unlike the East Coast PACT Protocol and the Canetti et al. Protocol, this protocol does not explicitly call for the use of network-level anonymization systems and does outline the proposed use of TLS to protect data in transit. Even with the use of network-layer anonymization systems, at a minimum, the seeds would be linkable to a geographical area with the use of health-care provider signatures.

While the protocol suggests the use of publishing delays, it is framed in a way to protect against replay attacks. However, the use of publishing delays would help in preventing adversaries from linking a series of records together. The protocol does make a mention that the process of appending records to an ever-growing list would further enable the linking of seeds and that randomizing the order of records could assist in this. The protocol concludes that without statistical analysis the benefits of this method are unknown.

6 TCN Protocol

The Temporary Contact Number (TCN) Protocol is a contact tracing protocol developed by the TCN Coalition [5]. The protocol aims to create a privacy-first contact tracing protocol, with trade-offs proposed to improve the scalability of the system.

6.1 Overview

The TCN Protocol white paper starts by proposing an ideal strawman where users broadcast random TCNs unlinkable to one another. While it is stated that unlinkable identifiers prevent many linkage attacks, the paper is quick to move on from this idea in a trade-off to improve the scalability of the system and the integrity of the records. The protocol suggests the use of public-private keys (referred to as Report Authorization Key (RAK) and Report Authorization Verification Key (RVK) respectively) where the first Temporary Contact Key (TCK) is generated by a hash of the private key and every subsequent TCK is generated by hashing the public key concatenated with the previous TCK. Using the TCK, the user generates a TCN by hashing together the 16-bit TCN number and the previous TCK. The TCN is then broadcasted over Bluetooth to neighboring devices. It is noted that the rotation of TCNs should correspond with the device's rotation of Bluetooth MAC address to prevent linkages. After the report rotation period, the user generates a new RAK and RVK pair.

In this protocol the user's device logs all of the used RAK and RVK pairs, in addition to the start and end times corresponding to when the keys were generated and retired. The user also keeps track of encountered TCNs from collocated devices. When an infected user has tested positive they upload individual records that consist of the concatenation of the RVK, the first TCK, the start time, end time, and a memo for implementation-specific details. The infected user then generates a signature for the message using their RAK so other users can verify the integrity of the message.

6.2 Evaluation

The coalition mentions that the ability for small-scale adversaries target and identify specific infected users by linking published records and the adversaries' logs is inevitable. As seen in the East Coast PACT Protocol, this can be defended against by allowing infected users to censor what they report to the registry. The protocol does not comment on whether infected users are allowed to do this.

To improve the scalability of the system, the protocol forgoes the use of unlinkable TCNs enabling large-scale adversaries with a network Bluetooth receivers to trace the travels of reported infected users with ease.

The protocol specifically mentions concerns of linking records together. To defend against adversaries attempting to link records in the public registry based on their time of upload, protocol implementations would allow infected users to upload individual records instead of requiring them to be uploaded in a batch.

The protocol does not comment on the use of network-level anonymization systems or the protocol's interaction with health-care providers, so the defense of these mechanisms could not be evaluated.

7 DP-3T Protocol

The Decentralized Privacy-Preserving Proximity Tracing (DP-3T) Protocol is a Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) endorsed protocol led from EPFL in Switzerland [6]. Similar to the other protocols, the project aims to enable the quick notification of users who have come in close contact with other infected users who have recently tested positive for COVID-19. The protocol outlines two potential designs for a system to achieve this goal; the Low-Cost Design and the Unlinkable Design.

7.1 Low-Cost Design Overview

In the Low-Cost Design, users generate a 32-byte secret key. This secret key is used to generate Bluetooth identifiers, referred to as ephemeral IDs (EphIDs), that are broadcasted periodically over a day. EphIDs are derived from the secret key using a PRF and passing the result into a PRNG. The result is then split into 16-byte chunks to obtain the EphIDs for the day. The protocol suggests the use of AES in counter mode for the PRNG and HMAC-SHA256 for the PRF. The next day, the user computes a cryptographic hash of the previous day's secret key to obtain a new key.

When a user is tested for the virus, they are provided with an authorization code that is activated after a positive result. After being notified of their positive test result by a health-care provider, the infected user is instructed to input the authorization code into their application to initiate the upload of the infected user's original secret key and corresponding day number. This pair is then broadcasted to all other users who use the key to locally generate the series of secret keys and corresponding EphIDs to compare to their list of encountered EphIDs.

7.2 Low-Cost Design Evaluation

Following the trend of other protocols, the Low-Cost Design opts for linkable tokens to increase the scalability of the system by reducing space complexity. Unlike other protocols, the Low-Cost Design uses tokens that are traceable for the entire duration of the infected user's sickness. This allows for a large-scale adversary to link every token from an infected user throughout their illness to physical locations through the use of a Bluetooth receiver

network with absolute certainty. Furthermore, the protocol does not defend infected users from small-scale adversaries attempting to link infection statuses.

As there is only ever one record uploaded per infected user, it does not make sense to evaluate the defense of linking individual records together.

It is stated that an eavesdropper on an infected user's network would be able to associate an upload to the backend server to an infected user's device. Despite this threat to the infected user, there is no mention of network-level anonymization systems as mentioned in the Canetti et al. Protocol and the East Coast PACT Protocol.

The protocol proposes the use of health-care provider issued authorization codes which by its nature allows uploaded records to be directly linked to a specific health-care provider.

7.3 Unlinkable Design Overview

The Unlinkable Design proposal enhances the Low-Cost Design to promote better privacy of infected users at the cost of scalability. In this design, the user generates a 32-byte seed that is then hashed and truncated into a 16-byte EphID. The EphID is broadcasted for a single time interval, at which point a new seed is generated and the method repeated. The user is required to store the series of 32-byte seeds used to generate the EphIDs, increasing the space requirement on the end-user. For each observed EphID, the user stores a hash of the EphID and timestamp, in addition to the proximity, duration, and coarse time used for future lookup.

When an infected user tests positive for COVID-19, they now have the option to redact identifiers by omitting certain time increments. To reduce the space complexity of unlinkable identifiers, the backend system creates a Cuckoo Filter that is then broadcasted to all users to compare against. A Cuckoo Filter is another probabilistic data structure used to test set membership, similar to a Bloom Filter. Cuckoo Filters differ from Bloom Filters in their underlying implementation, affecting insertion and lookup times. However, Cuckoo Filters can still only assert that values do not exist in a set or that they may exist in the set with a false positive probability.

7.4 Unlinkable Design Evaluation

The Unlinkable Design allows infected users to defend themselves against linkage attacks from small-scale adversaries by redacting records before upload, similar to the East Coast PACT Protocol. Given that the identifiers are no longer linkable to one another, a large-scale actor with a network of Bluetooth receivers would not be able to easily track the movements of an infected user. As mentioned in the evaluation of the Canetti et al. Protocol, this does not mean it is impossible and can be circumvented using similar methodologies used to circumvent Bluetooth MAC address randomization.

With the use of a Cuckoo Filter, only one record is uploaded per infected user so it does not make sense to evaluate the defense of linking individual records together.

The Unlinkable Design does not offer any additional security defending against network traffic analysis to link positive infection status to identities or against attacks attempting to link records to specific health-care providers.

8 Conclusion

In this survey of large scale decentralized contact tracing protocols, we have identified a number of common design patterns and trade-offs, and have evaluated their efficacy at mitigating linkage attacks. Common to all protocols was the minimization of risk encountered by negative users at the cost of increased risk for infected users who choose to disclose records with a central server. Protocols were evaluated in five ways:

- Vulnerable to small-scale adversaries linking records to infected user identities.
- Vulnerable to large-scale adversaries to linking Bluetooth identifiers with certainty.
- Vulnerable to individual records to be linked to one another.
- Vulnerable to using network traffic analysis to link positive infection status to identities.
- Vulnerable to linking records to specific health-care providers.

The high-level findings are summarized in **Figure 8.1**.

	Apple/ Google	**Canetti et al.**	**East Coast PACT**	**West Coast PACT**	**TCN**	**DP-3T (I)**	**DP-3T (II)**
Small-Scale Adversaries	Yes	Yes	Maybe	Yes	Yes	Yes	Maybe
Large-Scale Adversaries	Yes (Daily)	No	Yes (Hourly)	Yes (Daily)	Yes (Daily)	Yes	No
Individual Records	Unknown	Maybe	Maybe	Maybe	No	Not Applicable	Not Applicable
Network Traffic	Unknown	No	No	Unknown	Unknown	Yes	Yes
Health-care Providers	Unknown	No	Maybe	Yes (Signatures)	Unknown	Maybe	Maybe

Figure 8.1: A summary of the survey findings.

From the figure, it can be seen that most protocols opt to not protect infected users against small-scale adversaries. Those who attempted to defend them did so by allowing infected users to censor records before uploading, putting the onus on infected users. Fortunately, this design could be easily implemented into the other protocols by allowing users to censor selected records. This design choice comes at the cost of data reliability and availability. This could be mitigated by shortening the linkability period of records, reducing the potential overlap between times when infected users want to remain private (such as at home) and times when infected users believe their privacy is not at risk (e.g. public transit, concert halls).

In terms of protecting infected users' movements from being linked by large-scale adversaries, most protocols opted to allow this through the use of linkable identifiers. Using identifiers that are linkable for small time intervals allowed protocols to easily improve the scalability of the system. Protocols that attempted to protect infected users opted to not use linkable identifiers, but instead looked at implementing server-side filters to reduce space complexity.

In most protocols, the process of publishing individual records onto a central server was typically overlooked. In a naive implementation, infected user records could easily be linked to one another through the use of time analytics. The West Coast PACT Protocol and the TCN Protocol did attempt to mitigate this attack. In the West Coast PACT, it was proposed that the central server could implement publishing delays and it was noted that this would require a statistical analysis to prove it is an adequate defense mechanism. The TCN Protocol used a different design which allowed for infected users to upload records in partitions.

Methods of defending against the use of network traffic analysis to link positive infection status to identities were largely absent in the protocols surveyed. Fortunately, this could easily be defended with existing technologies, such as allowing connections from network-level

anonymization systems (such as Tor) as mentioned in the Canetti et al. Protocol and the East Coast PACT Protocol.

To prevent systems from being flooded with forged records, most protocols suggested that health-care providers would either authorize the upload of records or provide signatures to verify the authenticity and integrity of the records. Using either design leaves the protocol vulnerable to individual records being linked to specific health-care providers. The use of randomized authorization tokens by health-care providers, as suggested by the East Coast PACT Protocol, could adequately defend against this attack depending on the implementation. By its nature, the use of health-care provider signatures to verify records, as suggested by the West Coast PACT Protocol, would leave records vulnerable to this form of attack.

From the evaluation of the design choices and trade-offs, this survey reveals that it is possible to create a large scale decentralized contact tracing protocol that adequately defends infected users against linkage attacks. In general, these design choices come at the cost of scalability reliability of the system. When implementing a large scale decentralized contact tracing it is imperative that the consequences of these design choices be properly understood.

References

[1] Apple Inc. and Google LLC, "Contact Tracing - Cryptography Specification," *Apple Inc.*, Apr. 2020, Accessed: Apr. 19, 2020. [Online]. Available: `https://www.apple.com/covid19/contacttracing/`.

[2] R. Canetti, A. Trachtenberg, and M. Varia, "Anonymous Collocation Discovery: Harnessing Privacy to Tame the Coronavirus," *arXiv:2003.13670 [cs]*, Apr. 2020, Accessed: Apr. 19, 2020. [Online]. Available: `http://arxiv.org/abs/2003.13670`.

[3] R. Rivest *et al.*, "The PACT protocol specification," *PACT: Private Automated Contact Tracing*, Apr. 2020, Accessed: Apr. 19, 2020. [Online]. Available: `https://pact.mit.edu/`.

[4] J. Chan *et al.*, "PACT: Privacy Sensitive Protocols and Mechanisms for Mobile Contact Tracing," *arXiv:2004.03544 [cs]*, Apr. 2020, Accessed: Apr. 19, 2020. [Online]. Available: `http://arxiv.org/abs/2004.03544`.

[5] S. Niyogi *et al.*, "TCN Protocol," *TCN Coalition*, Apr. 2020, Accessed: Apr. 19, 2020. [Online]. Available: `https://github.com/TCNCoalition/TCN`.

[6] C. Troncoso *et al.*, "Decentralized Privacy-Preserving Proximity Tracing," *PEPP-PT*, Apr. 2020, Accessed: Apr. 19, 2020. [Online]. Available: `https://github.com/DP-3T/documents`.