PHyCT : Privacy preserving Hybrid Contact Tracing

Balancing centralised and decentralised models for privacy and tracing trade off

Mahabir Prasad Jhanwar[1] and Sumanta Sarkar[2]

[1]Ashoka University, INDIA, Mahavir.Jhawar@ashoka.edu.in
[2]TCS Innovation Labs, Hyderabad, INDIA, Sumanta.Sarkar1@tcs.com

Abstract

Ever since COVID-19 started grasping world's geographies one by one, countries have been struggling to tackle with this emergency by stretching their healthcare infrastructure beyond the boundary. World is now also trying to find ways to "live with the virus" or coping with the "new normal". In this effort, contact tracing is thought to be a vital tool which can quickly figure out persons that have come into vicinity of an infected person. Some countries have adopted centralized contact tracing in the perception that it is the most effective and easy solution. Centralized contact tracing has been in the centre of debate as it is a potential tool for launching mass surveillance. So objecting to this, decentralized model has been introduced which gives the control fully to the citizens. However, in decentralized model, the onus is completely on the users to act accordingly if they get a risk notification for coming in close contact with a COVID-19 positive patient. Decentralize model will fail if the large mass of users do not act accordingly after receiving the risk notification. Therefore, a balance needs to strike between the centralized and decentralized models given the socio-economic impact of this pandemic.

In this article, we take a hybrid approach and propose PHyCT that guarantees fail-safe, privacy, and security. This system acts like a decentralized one, where identities of users remain anonymous to the central authority. However, if there is a case of infection, the infected user and the central authority can *together only* reveal the identities of the users who have come in close contact. This feature enables to handle the situation if there are too many non-compliant users who do not report after getting infection exposure notification. Users who have not come into close contact of any infected person remain anonymous.

1 Introduction

Contact tracing is an important mitigation tool for national health services to fight epidemics such as COVID-19 . The goal of contact tracing is to quickly identify individuals who have been in close proximity to an infected person, before they even show symptoms, so that they may be tested, quarantined, and monitored for symptoms. Successful containment of the Coronavirus pandemic rests on the ability to remove such individuals from the circulating population.

Contact tracing begins once someone is confirmed as infected with a virus. The public health staff work with a patient to help them recall everyone with whom they have had close contact during the timeframe (usually 14 days for Coronavirus) while they may have been infectious. The health staff then alert these exposed individuals (contacts) of their potential exposure as quickly as possible. Regular follow-up are conducted with all contacts to monitor for symptoms and test for signs of infection. The manual in-person interview to trace contacts has the obvious limitation: patients don't remember who they have had contact with, or may not know the contact details of all those persons.

There is a growing interest in technology-enabled automated contact tracing systems to alleviate some of these problems. As smartphones are pervasive, so many countries have started using smartphone-based apps for quick and efficient contact tracing as part of the effort to manage the COVID-19 pandemic and prevent resurgences of the disease after the initial outbreak. These apps typically employ a combination of location services, such as the global positioning system (GPS), and neighbour discovery - the ability to discover phones in close proximity using Bluetooth [5].

There is a lack of privacy statement in the way many of these apps work. Some of these require each phone to send its current location to a central server every few seconds, and also continually send the server the numbers of the phones it had been in close proximity recently. Though, the contact tracing is rather obvious now, the system itself is clearly privacy-invasive. By continuously broadcasting the phone number of a user and her neighbours, it leaves an easy option to identify and track the person and her neighbours as well.

There are some recent initiatives to propose effective contact tracing protocols while guaranteeing privacy of personal information. There is also an ongoing debate on the underlying deployment framework of these, i.e. centralised versus decentralised. In the centralised architecture, personal data collected through the app is controlled by central authority. For the decentralised approach [7, 8], the personal data is enclosed or controlled by individuals only on personal devices. The later framework put an anonymous centralised database for only the infected people who have willfully disclosed their infection. With the announcement of the Apple and Google partnership to introduce decentralised contact-tracing functionality to iOS and Android [4], it seems likely that this will be adopted in many countries. On the same line a group of academics from Europe has also developed DP-3T [15]. A larger consortium known as Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) [1] has been formed which is working on both centralized and decentralized ones. INRIA of France and AISEC of Germany jointly working on a proposal, called ROBERT [11], for PEPP-PT initiative. One thing that is common among all these decentralised schemes is that central authority never learns the identity of the users infected or uninfected. Central authority only knows pseudoidentities of the contacts of an infected persons who came in the proximity in the recent past, broadcast the information to them.

After a user receives the information that she has come in a proximity of an infected person, the onus is completely on the user to test or self isolate/quarantine. However, if a large pool of users after getting the notification of a possible infection do not act accordingly, the purpose of contact tracing will fail. Perhaps, this is the reason that many countries still like to use centralized systems which gives the full control to the central authority to get the notification of possible infected users and identify them easily for faster containment of the infection spread.

Note that once a user is confirmed with infection and reports to the health department, then she willfully discloses her identity. Then it becomes an urgent task to identify the possible users who might have been infected by this user (second layer). More delay in catching infected users will cause further infection spread. Many people may hesitate to accept this disease and report accordingly in the fear of social abandonment, or simply may ignore the severity of this disease. If this class of people forms a large section of a society, then early containment of the disease will utterly fail. Thus no doubt there is a need to make a balance between privacy and protection, whereas protection comes at the cost of breaking anonymity. Ideally if the central authority gets to know the identities of the associated second layer of users once a user reports of infection, then it can act quickly.

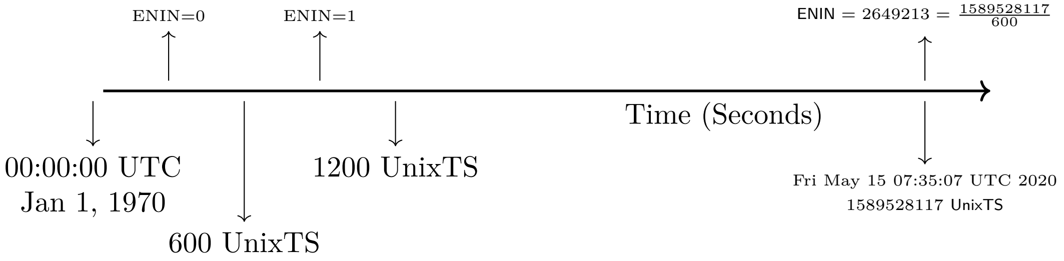
In this article, we take a hybrid approach. The central server keeps all the record of every user, and updates the users that are coming into proximity. But they don't really know the actual identities of all these users. Once a user reports of getting infection, then only with the help of this user, central server can break the anonymity of these second layer of users.

2 Preliminaries

In this section few basic concepts are discussed.

Definition 1 (Exposure Notification Interval Number (ENIN)[4]) *It is a unique positive integer representing a 10-minutes time-interval and is assigned to each timestamp in UNIX Epoch Time. For a UNIX "*Timestamp*", the corresponding* ENIN(Timestamp) *represents the unique 10-minute time-interval in which the* Timestamp *lies and is computed as follows:*

$$\mathsf{ENIN}(\mathsf{Timestamp}) = \lfloor \frac{\mathsf{Timestamp}}{600} \rfloor$$



Definition 2 (Pseudo Random Function [6]) *A family of functions* $\{F_s : \{0,1\}^k \to \{0,1\}^\ell \mid s \in \{0,1\}^n\}$ *is said to be pseudorandom if it satisfies the following two properties:*

1. *Easy to compute: The value* $F_s(x)$ *is easy to compute given any* $s \in \{0,1\}^n$ *and* $x \in \{0,1\}^k$

2. *Pseudorandom: The function* F_s *cannot be efficiently distinguished from a uniformly random function* $R : \{0,1\}^k \to \{0,1\}^\ell$*, given access to pairs* $(x_i, F_s(x_i))$*, where the* x_i*'s can be adaptively chosen by any probabilistic polynomial time distinguisher.*

3 Automated Contact Tracing

The existing Bluetooth-based automated contact tracing schemes are of two categories: centralized vs decentralized. A large number of systems, both centralized [2, 11] and decentralized [7, 8, 7, 13, 4] were proposed recently. The work in [16] showed the vulnerabilities and the advantages of both solutions systematically. In the following we briefly recall one system from both categories. Further the proposal from Apple and Google also has been analysed in [10].

3.1 Centralized Automated Contact Tracing

TraceTogether [2], deployed in Singapore, is a centralized Bluetooth-based contact tracing solution. The protocol is discussed in [9], and is presented here briefly. The system has the following components: Users holding a Bluetooth-equipped smartphone running the TraceTogether application, and a backend server acts as a repository for data to be pushed by smartphones upon authorisation by the respective users. The system works as follows:

- **Setup** Users get registered to the application and share their mobile numbers with the backend server.
- **Chirping** A user U generates a series of random ephemeral identifiers $\mathcal{U}_s = \{u_0, u_1, \ldots\}$, one for each time interval. The user's smartphone broadcast (every few seconds) the ephemeral identifier chosen for the current time interval. The user U also uploads the latest list $\mathcal{U}_s$ to the backend server.
- **Collision** At time t, users U and V encounter each other, exchanging u_k and v_k, where t falls in the kth time interval. Users keep the exchanged ephemeral identifiers with themselves. $\mathcal{U}_r$ denotes the set of identifiers received by U. Therefore, $v_k \in \mathcal{U}_r$.
- **Activate** If a user U is diagnosed positive, she must upload the final received list $\mathcal{U}_r$ to the "exposure database" on the backend server.
- **Alert** Every time an infected user U uploads her encounter history $\mathcal{U}_r$ to the exposure data base, the backend server must ensure that an alert is sent out to all contacts of U who came in her close proximity in the recent time. The server matches the membership of each identifier from $\mathcal{U}_r$ to ephemeral identifier sets for all mobile numbers, and sends out an alert to the matching mobile number accordingly. For example, the identifier $v_k \in \mathcal{U}_r$ will be a member of V_s registered against the mobile number of V, and therefore an advisory alert will be issued to V.

3.2 Decentralized Automated Contact Tracing

We now describe the PACT (Private Automated Contact Tracing) protocol [13], a simple, decentralized approach to using smartphones for contact tracing based on Bluetooth proximity. Users of this scheme do not reveal anything about themselves, unless they volunteer to do so. The system has the following components: Users holding a Bluetooth-equipped smartphone running the application, health staff and a backend server acts as a repository for data to be pushed by smartphones upon authorisation by the health staff. The system works as follows:

- **Setup** Users get registered to the application. The application uses a pseudo random function F.
- **Chirping** A user U generates a series of random seeds $\mathcal{U}_s = \{u_0, u_1, \ldots\}$, one for each time interval. The user's smartphone broadcast every few seconds an ephemeral identifier as follows. At time t, falling in the time interval say k, the ephemeral identifier broadcasted is $u_{kt} = F(u_k, t)$.
- **Collision** At time t, users U and V encounter each other, exchanging u_{kt} and v_{kt}, where t falls in the kth time interval. Users keep the exchanged ephemeral identifiers with themselves. Let $\mathcal{U}_r$ denotes the set of identifiers received by U. Therefore, $v_{kt} \in \mathcal{U}_r$.

- **Activate** If a user U is diagnosed positive, she must upload her seeds $\mathcal{U}_s$ to the "exposure database" on the backend server upon authorisation by a health staff.
- **Alert** Every time an infected user U uploads her seeds $\mathcal{U}_s$ to the exposure data base, all contacts of U who came in her close proximity in the recent time will have a way to know about this. Users of the scheme periodically check the identifiers collected by their phones against the downloaded exposure data base from the backend server. A user V finds out if she was exposed to an infected individual in the recent time as follows.

```
For every seed $s$ in Exposure Database
    For every time-instance $t$ in time-interval designated for the seed $s$
        If $F(s,t) \in \mathcal{V}_r$
            Alert: Exposed
Alert: Safe
```

3.3 Privacy and Security

A reasonable framework to model privacy for automated contact tracing system must address what information about which user type should be hidden from whom [3, 12, 16]. These dimensions can be introduced by discussing the three typical user types, the attacker models, and the ways how privacy can be impacted.

1. Types of Users: In the following a distinction is made amongst the active users of an automated contact tracing system.
   (a) Normal: This type referred to those users who have not reported positive and not identified as at-risk.
   (b) At-risk: This type referred to those users who have been alerted by the system for being in the close vicinity of an infected individual in recent time.
   (c) Infected: This type referred to those users who have been identified as infected with the virus.
2. Types of Attackers: We now consider the types of attackers against which the system must preserve the users' privacy.
   (a) Malicious Application Users: They represent both passive and active users of the application.
   (b) Administrator. This type of attacker has access to partial/entire data-base of users and their shared information. They represent various entities including administrators at various levels and privileged users such as health staffs and government officials.
   (c) Third Parties: This type is referred to those attackers who can eavesdrop, modify, and suppress any message that is sent over a public network.

The goal of the attackers is to collect, process, or transmit any more data than what is necessary to achieve the purpose of supporting public health measures for the containment of the virus. The goal is also to de-anonymize user's identity by mapping the user's pseudonym to her real identity, or narrowing down on the real identity of a user without necessarily identifying her individually - such as place of living.

The security of an automated contact tracing system must make it difficult for third parties to fake ”*at-risk*” notifications to normal users. Such false notifications would adversely affect normal users, and consequently overwhelm a country's healthcare system with the unnecessary quarantine measures. Fake *at-risk* notifications could easily be generated as a result of the system failing to ensure that only users who are really infected are handled by the system as such. The false reporting could also pollute data used for epidemiological research. The security for the existing systems currently assumes that the backend server is trusted to not add or remove information shared by the users and to be available.

4 Automated Hybrid Contact Tracing

We now describe our hybrid contact tracing scheme. The system has the following components:

- Users holding a Bluetooth-equipped smartphone running the application,
- health staff,
- a backend server T_{E} acts as a repository for exposure data to be pushed by infected user's smartphones upon authorisation by the health staff, and finally
- a central authority managed backend server T_{CA} acts as a repository for encrypted data to be pushed by regular users every day.

The system works as follows:

- **Setup** Users get registered to the application. The cryptographic algorithms used by the applications are:
  - Pseudo random function F
  - 2-out-of-2 secret sharing $\Pi = (\Pi.\mathsf{Share}, \Pi.\mathsf{Recon})$ [1]
  - Hash function H
  - Symmetric key encryption $\mathcal{E} = (\mathcal{E}.\mathsf{Enc}, \mathcal{E}.\mathsf{Dec})$
- **Chirping** The mobile application of every user broadcasts ephemeral identifiers to its peripheral devices at a regular interval. It also uploads encrypted data to a central authority once every day. The chirping details are as follows:
  1. Every 10 minutes, the user's mobile application generates a seed s_e, where the suffix e is the unique exposure notification interval number ENIN associated to the above 10 minute interval. These seeds are kept in the local storage of the mobile, and for a user U, let $\mathcal{U}_s$ denote the set of all locally generated and saved seeds.
  2. Every few seconds, the mobile does the following. For a given Unix time stamp t, it computes $e_t = \lfloor \frac{t}{600} \rfloor$ which represents the unique ENIN e in which t falls (consequently, $s_{e_t} = s_e$).

     Next it computes a message tuple $(m_t = (m_{t1}, m_{t2}), m_{e_t} = (c_1, c_2))$ at time t as follows:

     (a) Compute $(s^B_{e_t}, s^C_{e_t}) \leftarrow \Pi.\mathsf{Share}(s_{e_t})$. ($B$ represents *broadcast*, and C represents the *central authority*)

     (b) Compute $\mathsf{m}_{t1} = F(s_{e_t}, t)$, and $m_{t2} = s^B_{e_t}$

     (c) Compute $c_1 = \mathcal{E}.\mathsf{Enc}_{s_{e_t}}(\mathsf{id}_U)$, and $c_2 = s^C_{e_t}$

     Finally,

[1] See [14] for the details about secret sharing.

- It broadcasts $m_t = (m_{t1}, m_{t2})$ to its peripheral devices, and
- Uploads $m_{e_t} = (c_1, c_2, H(s^B_{e_t}))$ to the central authority C.

Note that for two different t_1, t_2 falling into the same ENIN, we have $e_{t_1} = e_{e_2}$, therefore $s_{e_{t_1}} = s_{e_{t_2}}$, and consequently $m_{e_{t_1}} = m_{e_{t_2}}$. The user can choose to store all m_e's for the day and uploads them to T_{CA} together at the end of the day.

- **Collision** A collision happens when two users, say $\mathsf{U}_1, \mathsf{U}_2$, are in a closer proximity and as a result their mobiles receive broadcasted messages from each other as follows.

  - U_1 receives $\mathsf{m}_{t_2}^{B_{U_2}} = (F(s^{U_2}_{e_{t_2}}, t_2), s_{e_{t_2}}^{B_{U_2}})$ from U_2
  - U_2 receives $\mathsf{m}_{t_1}^{B_{U_1}} = (F(s^{U_1}_{e_{t_1}}, t_1), s_{e_{t_1}}^{B_{U_1}})$ from U_1,

  where t_1, t_2 differs by a few seconds, and potentially falls into the same ENIN, and $s^{U_i}_{e_{t_i}}$ is the seed U_i generated for the ENIN e_{t_i}. Users store the received messages in their local storages. For a user U, let $\mathcal{U}_r$ denote the set of all collected chirps other users.

- **Activate:** In the activation step, a newly infected user U uploads, upon authorisation by the health staff, all her seeds to the exposure data base on the server T_{E}, i.e., all pairs $\{(e, s_e)\}$ for all unique ENIN e's falling between the window of U's registration and the current event of U being notified as a Covid positive.

  U also uploads, upon authorisation by the health staff, all collected chirp-message tuples $\{m^{\mathsf{V}_1}, m^{\mathsf{V}_2}, \ldots, m^{\mathsf{V}_k}\}$ that it has received over time period which is considered its contagion period to the central authority server T_{CA}. Any such m^{V_i} represents a tuple $(F(s^{V_i}_{e_t}, t), s_{e_t}^{B_{V_i}})$, received from a user V_i at time instance t during a close proximity interaction.

- **Alert:** Every time an infected user U uploads all her seeds $\mathcal{U}_s$ and all collected chirps $\mathcal{U}_r$ to the exposure data base, all contacts of U who came in her close proximity in the recent time will have a way to know about this. Users of the system periodically check the chirps collected by their phones against the downloaded exposure data base from the backend server. A user V finds out if she was exposed to an infected individual in the recent time as follows. It must download seeds from exposure data base, uploaded by infected individuals. Assume, for a specific ENIN e, the downloaded seeds are $\{s_e^{\mathsf{U}_1}, s_e^{\mathsf{U}_2}, \ldots, s_e^{\mathsf{U}_n}\}$. The ENIN e defines a 10 minute window. The user V must run the following loop to check if there is a match:

```
For $k$ in $1 \leq k \leq n$
    For $t$ in $1 \leq t \leq 600$
        If $F(s_e^{\mathsf{U}_k}, t) \in \mathcal{V}_r$
            Alert: Exposed
Alert: Safe
```

- **Report:** If Exposed, the user complies to report to health department.

- **Trace:** In this step an exposed but non-compliant user can be traced by the Central Authority as follows. Assume V is a non-compliant user who recently came in contact with an infected user U. The central authority must be able to trace V's identity with the help of $\mathcal{U}_r$ (U's collected chirps) and the data that V has shared with the central authority till date. Assume the close-proximity interaction between U and V took place during a time stamp t and let e_t is the corresponding ENIN. The data that the central authority can use to uncover U's identity comprises of

- the broadcasted chirp $(F(s_{e_t}, t), s^B_{e_t})$ that U received from V, and
- the data $\left(c_1 = \mathcal{E}.\mathsf{Enc}(\mathsf{id}_\mathsf{V}, s_{e_t}), c_2 = s^C_{e_t}, H(s^B_{e_t})\right)$ that V shared with the central authority on that day.

The hash $H(s^B_{e_t})$ helps the central authority to match $c_2 = s^C_{e_t}$ and $s^B_{e_t}$ as shares of the seed s_{e_t}. It runs $\Pi.\mathsf{Recon}(s^C_{e_t}, s^B_{e_t})$ to compute s_{e_t}.

Finally, V's identity is uncovered by computing $\mathcal{E}.\mathsf{Dec}(c_1, s_{e_t})$.

Additionally, robustness of tracing is also guaranteed by checking if

$$F(\mathsf{reconstructed\ secret}, t) \overset{?}{=} F(s_{e_t}, t).$$

4.1 Features: Fail-safe, Privacy and Security

Our system provides an additional layer on to a typical distributed contact tracing scheme such as PACT [13]. This additional layer has helped the system become fail-safe. **Trace** is designed in such a way that the central authority will be able to trace the identity of individuals who received virus-exposure notification but are non-compliant in reporting it to the health staff. However, this is only possible once an infected user provides the shares that she has received from her contacts. This provides the privacy guarantee as explained below.

The application users, in addition to broadcast chirp messages to its peripheral devices, also shares information with the central authority. Similar to the PACT scheme, the primary component of our chirp message is an ephemeral identifier $F(s_e, t)$ computed using the secret seed s_e. The additional chirp component is a share s^B_e of s_e. Security of a 2-out-of-2 secret sharing implies that having access to at most one share s^B_e does not reveal any information on the secret; neither the central authority nor the user can get the identities alone. Each element from the seed space remains an equally likely candidate for the secret. This means, our broadcast step does not present an additional advantage to a malicious attacker compared to the broadcast step of the PACT scheme. The users in our systems also share with the central authority their encrypted identities. In each sharing to the central authority, a user's real identity is encrypted $\mathcal{E}.\mathsf{Enc}(\mathsf{id}_U, s_e)$ under a fresh seed s_e and the same is uploaded along with a share s^C_e of s_e. In the absence of the other share, no information is revealed on the secret seed and consequently the identity id_U to the central authority. Therefore, the central authority on its own cannot uncover user identities.

5 Conclusions

We have presented PHyCT which is a hybrid approach for contact tracing of $\mathsf{COVID\text{-}19}$ suspects. Since this is not a typical centralized model, users feel more comfortable with this model, at the same time central authority has all the scope of knowing the identities of the contacts of a user who is confirmed with infection. So we have been able to keep the balance between the users' privacy and the necessity of early detection. We believe that this approach will be very much useful in dealing with the new normal and containing the pandemic.

References

[1] Pan-European Privacy-Preserving Proximity Tracing. https://www.pepp-pt.org/.

[2] Singapore Government Technology Agency. TraceTogether app. https://www.tracetogether.gov.sg/,. Released March 21, 2020.

[3] Fraunhofer AISEC. Pandemic contact tracing apps: Dp-3t, pepp-pt ntk, and robert from a privacy perspective. Cryptology ePrint Archive, Report 2020/489, 2020. https://eprint.iacr.org/2020/489.

[4] Apple and Google. Privacy-preserving contact tracing. https://www.apple.com/covid19/contacttracing.

[5] Bluetooth. Bluetooth is moving proximity solutions in the right direction. https://www.bluetooth.com/blog/bluetooth-proximity-solutions/.

[6] Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 79–158. Springer International Publishing, 2017.

[7] Ran Canetti, Ari Trachtenberg, and Mayank Varia. Anonymous collocation discovery: Taming the coronavirus while preserving privacy. *CoRR*, abs/2003.13670, 2020.

[8] Justin Chan, Dean P. Foster, Shyam Gollakota, Eric Horvitz, Joseph Jaeger, Sham M. Kakade, Tadayoshi Kohno, John Langford, Jonathan Larson, Sudheesh Singanamalla, Jacob E. Sunshine, and Stefano Tessaro. PACT: privacy sensitive protocols and mechanisms for mobile contact tracing. *CoRR*, abs/2004.03544, 2020.

[9] Hyunghoon Cho, Daphne Ippolito, and Yun William Yu. Contact tracing mobile apps for COVID-19: privacy considerations and related trade-offs. *CoRR*, abs/2003.11511, 2020.

[10] Yaron Gvili. Security analysis of the covid-19 contact tracing specifications by apple inc. and google inc. Cryptology ePrint Archive, Report 2020/428, 2020. https://eprint.iacr.org/2020/428.

[11] France INRIA and Germany AISEC. Robert: Robust and privacy-preserving proximity tracing. https://github.com/ROBERT-proximity-tracing/documents/blob/master/ROBERT-summary-EN.pdf.

[12] Christiane Kuhn, Martin Beck, and Thorsten Strufe. Covid notions: Towards formal definitions - and documented understanding - of privacy goals and claimed protection in proximity-tracing services. https://arxiv.org/abs/2004.07723. *CoRR*, abs/2004.07723, 2020.

[13] MIT. Pact: Private automated contact tracing. https://pact.mit.edu/.

[14] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[15] Carmela Troncoso, Mathias Payer, Jean-Pierre Hubaux, Marcel Salathé, James Larus, Edouard Bugnion, Theresa Stadler Wouter Lueks, Apostolos Pyrgelis, Sylvain Chatel Daniele Antonioli, Ludovic Barman, Kenneth Paterson, Srdjan Capkun, David Basin, Jan Beutel, Dennis Jackson, Bart Preneel, Nigel Smart, Dave Singelee, Aysajan Abidin, Seda Gürses, Michael Veale, Cas Cremers, Michael Backes, Nils Ole Tippenhauer, Reuben Binns,

Ciro Cattuto, Alain Barrat, Giuseppe Persiano, Dario Fiore, Manuel Barbosa, and Dan Boneh. Decentralized privacy-preserving proximity tracing. https://github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf.

[16] Serge Vaudenay. Centralized or decentralized? the contact tracing dilemma. *IACR Cryptol. ePrint Arch.*, 2020:531, 2020.