Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System[*]

Gennaro Avitabile[1], Vincenzo Botta[1], Vincenzo Iovino[1], and Ivan Visconti[1]

[1]DIEM (S^3 Lab.), University of Salerno, Italy,
{gavitabile,vbotta,viovino,visconti}@unisa.it

April 27, 2020

Abstract

Mass surveillance can be more easily achieved leveraging fear and desire of the population to feel protected while affected by devastating events. In such cases governments are more legitimate to adopt exceptional measures that limit civil rights, usually receiving large support from their citizens.

The COVID-19 pandemic is currently affecting the freedom and life style of many citizens in the world. People are forced to stay home for several weeks, unemployment rates quickly increase, uncertainty and sadness generate an impelling desire to join any government effort in order to stop as soon as possible the spread of the virus.

Following recommendations of epidemiologists, governments are proposing the use of smartphone applications to allow automatic contact tracing of citizens. Such systems can be an effective way to defeat the spread of the SARS-CoV-2 virus since they allow to gain time in identifying potentially new infected persons that should therefore be in quarantine. This raises the natural question of whether this form of automatic contact tracing can be a subtle weapon for governments to violate the privacy of their citizens as part of new and more sophisticated mass surveillance programs.

In order to preserve privacy and at the same time to contribute to the containment of the pandemic, several research partnerships are proposing privacy-preserving contact-tracing systems where pseudonyms are updated periodically to avoid linkability attacks. A core component of such systems is Bluetooth low energy (BLE, for short) a technology that allows two smartphones to detect that they are in close proximity. Among such systems there are some proposals like DP-3T, PACT and the Apple&Google exposure notification system that through a decentralized approach guarantee better privacy properties compared to other centralized approaches (e.g., PEPP-PT-NTK, PEPP-PT-ROBERT). On the other hand, advocates of centralized approaches claim that centralization gives to epidemiologists more useful data, therefore allowing to take more effective actions to defeat the virus.

Motivated by Snowden's revelations about previous attempts of governments to realize mass surveillance programs, in this paper we first analyze mass surveillance attacks that leverage

[*] Disclaimer: this work is based on our understanding of all sources of information specified in the bibliography. New relevant documents and revisions of previous documents appear on-line on a daily basis. Not everything is clear to us and thus we ask in the Introduction several natural questions. In case we have misunderstood something or the answers to our questions are known already, we would be happy to be notified and then we will promptly make proper updates.

weaknesses of automatic contact systems. We focus in particular on the DP-3T system (still our analysis is significant also for PACT and Apple&Google systems) that has been endorsed by Apple&Google. The endorsement has the impact of integrating in the forthcoming update of Android and iOS special features like a synchronous rotation of the BLE MAC address of the smartphone with the update of the pseudonyms of the DP-3T system.

Based on recent literature and new findings, we discuss how a government can exploit the use of DP-3T to successfully mount privacy attacks as part of a mass surveillance program.

Interestingly, we also show that the privacy issues in DP-3T are not intrinsic in any BLE-based contact tracing system. Indeed, we propose a different system named Pronto-C2 that, in our view, enjoys a much better resilience with respect to mass surveillance attacks still relying on BLE. Pronto-C2 is based on a paradigm shift: instead of asking smartphones to send keys to the Big Brother (this corresponds to the approach of DP-3T), we construct a decentralized BLE-based ACT system where smartphones anonymously and confidentially talk to each other in the presence of the Big Brother.

Pronto-C2 can optionally be implemented using Blockchain technology, offering complete transparency and resilience through full decentralization, therefore being more appealing for citizens. Only through a large participation of citizens contact-tracing systems can be very useful to defeat COVID-19, and our proposal goes straight in this direction.

Contents

1 Introduction

In 2013 Edward Snowden disclosed global surveillance programs [CHRT20] opening a worldwide discussion about the tradeoff between individual privacy and collective security.

Uncertainty and fear may strongly affect citizens' psychology. Public dangers like crimes, terrorism and natural disasters can be an excuse used by governments to set up mass surveillance program with the actual goal of controlling the population.

SARS-CoV-2. A major threat is currently affecting humanity: the COVID-19 pandemic. The aggressiveness and fast spread of the SARS-CoV-2 virus have a strong impact on public opinion. Several governments are taking the most restrictive measures of the last decades in order to contain the loss of human lives and to preserve their economies. Fear is spreading, citizens are forced to stay home, many jobs have been lost, and more dramatically the number of deaths goes up very fast day by day.

Automatic contact tracing. According to epidemiologists, a major problem is that the virus spreads very quickly while current procedures to detect infected people and to find and inform potentially infected people are slow. When a new infected person is detected, too much time is spent to inform her recent contacts and to take proper restrictive actions. Commonly by the time when some of the recent contacts are informed and are put in a quarantine, they might have already infected others.

In order to improve current systems many researchers are proposing automatic systems for contact tracing. Such systems can dramatically increase chances that recent contacts of an infected person are informed before infecting others. Essentially, whenever a person is diagnosed as infected, immediately all her recent contacts (i.e., persons that have been in close proximity to the infected one) are informed. This allows to promptly take appropriate countermeasures.

Automatic contact tracing (ACT, for short) is therefore considered an important component that in synergy with physical distancing and other already existing practices can contribute to defeating the SARS-CoV-2 virus.

Privacy threats. There are serious risks that such systems might heavily affect privacy. Citizens could be permanently traced and arguments like "If you have nothing to hide, you have nothing to fear" (Joseph Goebbels - Reich Minister of Propaganda of Nazi Germany from 1933 to 1945) are already circulating in social networks. Governments could leverage the world-wide fear to establish automatic contact tracing systems in order to realize mass surveillance programs.

Motivated by such risks, several researchers and institutions are advertising to citizens the possibility of realizing automatic contract tracing systems that also preserve privacy to some extent. Such systems crucially rely on Bluetooth low energy (BLE, for short).

The BLE-based approach. BLE is a technology that allows smartphones physically close to each other to exchange identifiers requiring an extremely low battery consumption. Such communication mechanism avoids GPS technology and third-party devices like Wi-Fi routers or base stations of cellular networks. It is therefore a viable technology to allow the design of automatic privacy-preserving contact-tracing systems.

BLE-based tracing is used by Apple in their privacy-preserving system to allow to find lost devices [Gre19]. Matthew Green in a interesting webinar with Yehuda Lindell [GL20] explicitly proposed to start with Apple's tracking system when trying to design a privacy-preserving proximity contact-tracing system for citizens. Apple and Google have very recently announced a partnership to provide an application program interface for exposure notification (AGEN, for short) [AG20] that can be used to include such features in smartphone applications.

In parallel with the Apple&Google initiative, other BLE-based approaches very similar in spirit have been integrated in automatic contact-tracing systems and are currently used or about to be used in many countries. Such BLE-based systems commonly rely on the use of pseudonyms that smartphones announce through BLE beacons. After a short period each smartphone replaces the already announced pseudonym with a (seemingly independent) new one. Each smartphone receives pseudonyms sent by others and stores them locally. Therefore a smartphone will have a database of the announced pseudonyms and a database of the received pseudonyms. The central idea is that whenever a person is detected infected then smartphones that have been physically close to the smartphone of the infected person should be notified and should compute a local risk scoring. In order to realize this, the smartphone of the infected person should use the above two databases to somehow reach out the smartphones that have recently been physically close to it. This communication is achieved through a backend server as follows. First the smartphone of the infected person will use the above two databases to communicate data to the backend server. The server could run some computations on data received from smartphones of infected citizens. The server will also use collected/computed data to answer pull requests of smartphones that desire to check if there is any notification for them.

Intuitively, such approaches through the unlinkability of the pseudonyms guarantee some degree of privacy. Despite the privacy-preserving nature of the BLE-based approach, the risk that such systems can be misused to realize mass surveillance programs remains a major concern that might slowdown the actual adoption of such systems. Indeed, most governments will not impose their use, leaving to citizens the option to decide[1].

Centralized vs Decentralized BLE-Based ACT. An important point of the design of a BLE-based ACT system is the generation of pseudonyms used by smartphones. Two major approaches have been proposed so far.

In a centralized approach pseudonyms are generated by the server. Each smartphone, during the setup of the ACT smartphone application connects to the server and receives its pseudonyms. Therefore the server knows all the pseudonyms honestly used in the system. This is pretty obviously a clear open door to mass surveillance. Such dangers are discussed in [DT20a]. Currently the centralized approach is part of the protocols named NTK and ROBERT that are developed inside the Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) initiative [PEP20].

The decentralized approach breaks the obvious linkability of pseudonyms belonging to the same smartphone by letting the smartphone itself generate such pseudonyms.

Straight-forward decentralized BLE-Based ACT. The most trivial way to realize a decentralized BLE-Based ACT system consists of giving to the server the role of proxy that forwards to non-infected persons the pseudonyms of those infected persons that decide to upload their pseudonyms after being detected infected. Therefore everyone, including the server, clearly learns directly pseudonyms that have been used during the previous days by a recently infected persons. Instead the pseudonyms generated by smartphones belonging to non-infected persons are not uploaded to the server. Such pseudonyms remain visible only to whoever was physically close to those

[1] There is an explicit recommendation of the EU commission [Com20] towards leaving optional the use of such systems in addition to make sure that privacy is preserved.

smartphones. In terms of privacy, such straight-forward decentralized systems seemingly have a potential to offer a better protection compared to known systems that use the centralized approach. There are a few proposals based on the straight-forward decentralized approach, most notably Decentralized Privacy-Preserving Proximity Tracing (DP-3T, for short) and Private Automated Contact Tracing (PACT, for short).

Is privacy-preserving ACT a fig leaf? The unlinkability of pseudonyms advertised in BLE beacons is completely useless if the BLE MAC address associated to a smartphone does not change in a synchronized way with the pseudonyms [BLS19]. Notice that iOS and Android are (almost completely) the currently deployed operating systems for smartphones and have some serious restrictions on updating of BLE MAC address. The smartphone application should obviously work in the background and should have control over its BLE MAC address so that this value can rotate along with the pseudonyms announced in the BLE beacons. This contrasts with the above restrictions. Therefore it is technically hard (if not impossible) to realize privacy-preserving smartphone applications using BLE.

The move of Apple&Google. Interestingly, Apple &Google are promising forthcoming updates for iOS and Android providing AGES at operating system level[2] resolving along with it also the MAC address linkability problem. However the two features are seemingly connected, more precisely: if you want to design a smartphone application that needs to rotate the BLE MAC address synchronously with the content of the BLE beacon then you must use their API and therefore you must use their approach for pseudonym generation and exposition.

This lack of flexibility generates some interesting consequences. First of all, the centralized approach does not seem to be implementable since it relies on pseudonyms generated by the server and then advertised in the BLE beacon by the smartphone. However the generation of pseudonyms can only happen inside the smartphone when using AGES. Such mismatch seems to imply that the decision of Apple &Google will exclude the centralized approach to privacy-preserving ACT, making non-applicable some decisions of some governments that have a bias towards centralization.

In Italy the government assigned to the company "Bending Spoons" the task of realizing a centralized privacy-preserving ACT named "Immuni"[ds]. While the company was initially part of PEPP-PT, "Bending Spoons" has very recently decided to switch to the decentralized approach using AGES. This motivates the following natural question. *Q1: Is the change from the centralized to the decentralized approach against the will of the Italian government and therefore forced by Apple&Google's decision to offer support only for systems compatible with AGES? More in general, are we fine with being forced to choose the contact-tracing approach decided by Apple&Google?*

Snowden's revelations included memos confirming the existence of backdoors (e.g., see Dual_EC_DRBG) in standardized cryptographic algorithms [Wik]. Above doubts therefore motivate the following natural question. *Q2: Can we exclude that the system proposed by Apple&Google will not be abused to realize mass surveillance programs?*

[2] They will provide only the part concerning the generation, rotation, and exposure of pseudonyms along with a flag to activate/dis-activate this service in the settings. There will not be user applications and neither a server collecting pseudonyms.

1.1 Our Contribution

Starting with the inspiring list of attacks presented by Vaudenay [Vau20] and taking into account the answers given by DP-3T in [DT20c], in this work we first analyze the degree of privacy protection achieved by DP-3T with respect to mass surveillance attacks. In such attacks a government through its natural power controls (even partially) the server, the laboratories that check infections and the national territory to realize mass surveillance programs.

We consider quite dangerous the fact that in DP-3T (and all analogue systems) one can be traced even when walking alone, silently. Indeed, a passive antenna can detect the pseudonym without transmitting anything, and can later on check if it belongs to the list of infected persons. It is easy to link the real identity of an infected person with the pseudonyms she used in the last two weeks. Indeed, such antennas can also be nearby laboratories where one is tested to check infection and this allows to connect pseudonyms to identity. We believe that this is an open door to mass surveillance and one should instead focus on privacy-preserving systems where silent tracing attempts are ineffective. Also other BLE devices that are in general used for other purposes (e.g., information kiosks) can be used to trace people. Obviously one can not expect that nothing else will be done with BLE except contact tracing, and thus preserving privacy while other uses of BLE continue is a necessary goal. Notice also that the use of active kiosks running precisely the BLE-based contract-tracing protocol is actually recommended in [Tea20] (see Remark II in Section 3.2). Instead, we believe that they can be a source for privacy attacks. The lack of privacy with respect to such adversaries is a major thorn in the side of DP-3T and other analogue systems[3]. We stress that the issues exist regardless of the update of the MAC address of the BLE device. Technically speaking, the key weakness of DP-3T is actually a weakness of the straight-forward decentralized approach: asking smartphone applications to hand over the used keys/pseudonyms to the server is like asking citizens to kneel down in front of the Big Brother.

Recently several scientists mainly specialized in cryptography and information security have signed a joint statement [doc20] on contact tracing to state that when multiple possible options are possible one should select the most privacy-preserving solution (as long as it is as effective as others). The decision of Apple&Google is in complete contrast with the above statement since it does not allow to choose among different options and could penalize options that offer more privacy. This motivate the following natural question. *Q3: Since Apple&Google are seemingly excluding the implementation of decentralized privacy-preserving ACT systems that do not follow the straight-forward approach, will the same community of scientists ask Apple&Google to release an update iOS and Android in order to allow governments to wisely choose the most privacy-preserving solutions (as specified in [doc20])?*

Next we present Pronto-C2, a new decentralized privacy-preserving automatic proximity contact tracing system based on BLE. We show that our system is arguably more resilient than others against mass surveillance attacks, while remaining useful for epidemiologists. Our system can be implemented through government servers but also can be fully decentralized using blockchain technology. We believe that full decentralization can play an important role to help the work of epidemiologists since citizens are certainly more prone to use a smartphone contact-tracing application in a system that is transparent and resilient to attacks, in addition to being privacy preserving.

[3] We will instead show that our Pronto-C2 system does not suffer of such drawbacks, see the paragraphs entitled "Silent tracing" and "Shameless tracing".

1.2 High-Level Overview of Pronto-C2

Our solution can be seen as a paradigm shift compared to the straight-forward decentralized approach. Indeed instead of asking infected people to hand over their keys to the Big Brother, we allow citizens to anonymously and confidentially call each other in the presence of the Big Brother. The way we do it is explained below.

In the 70s Merkle, Diffie and Hellman invented public-key cryptography. Starting with Merkle's puzzles, Diffie and Hellman proposed a key exchange protocol [DH76] (i.e., the Diffie-Hellman protocol) where two parties can establish a secret key K by just sending one message each on a public channel. A message consists of a group element in a setting where the so called Decision Diffie-Hellman assumption holds.

In our view, the most natural way to realize a privacy-preserving ACT system consists of having as pseudonym a group element that corresponds to a message in the DH protocol. This natural idea was also proposed to the DP-3T team by the github user a8x9 [a8x]. In order to actually realize such form of ACT system, one needs to solve the following two main problems.

Anonymous call: realizing a mechanism that allows an infected party to transfer K to the other party in a secure and privacy-preserving way.

Shortening pseudonyms: making sure that the size of a group element fits the number of available bits in a BLE beacon.

Calling (anonymously) the infected person. We solve the first problem by asking the infected party, after having received a proper authorization from the laboratory that detected the infection, to upload K along with the authorization to a bulletin board. The bulletin board can be just managed by a server as in DP-3T, but we actually suggest to implement the bulletin board with a decentralized blockchain so that we can decentralize the server making the entire process transparent and reliable.

When implementing this step using a blockchain, the verification of the authorization must be performed by a smart contract and thus the check should be accomplished uniquely with public information. For this reason, we suggest the use of digital signatures. In order to make unlinkable the upload of K with the infected party, we suggest the use of blind signatures [Cha83].

In this work when referring generically to a blockchain we always mean a permissioned blockchain. Possibly, the governance should be decentralized (e.g., Hyperledger Fabric [ABB$^+$18]). If performance issues require to use a centralized server, then we insist that all data should remain public without leaving any specific secret to the server. Moreover the server should periodically (e.g., every 10 minutes) notarize on the Bitcoin blockchain the cryptographic hash of the new data arrived in last time interval. This will show public evidence of cheating in case there is any fraud on the server, and citizens can obviously switch off the application since a mass surveillance attack might be in progress.

Notice that our approach is therefore completely different from DP-3T. Indeed while in DP-3T the pseudonyms of the infected party are sent in broadcast to everyone (or added to a Cuckoo filter by the server that then transmits the filter) we instead ask the infected party to send a message that is understandable uniquely by the party with which she was in close proximity. Therefore K is more like a phone call where the infected party sends to the answering party the following message[4]

[4] The Italian word "Pronto" stays for "Hello" and C2 pronounced in English stays for "it is you" in Neapolitan language as in the title of a very popular song of Nino D'Angelo [D'A83] (see also the movie [Lau83], min. 59:00).

"Hello, it is you that were next to me... and I've just discovered that I'm infected".

Every person that is not infected will connect to the server (or to the blockchain) and will check all published keys, looking for K. Notice that there is a different key K to check for every BLE beacon received in the last two weeks. The entire search process can be very fast by having servers that also keep a sorted version of all keys so that anyone that would like to check the existence of a given K can just perform an on-line binary search. Such servers can still be run by the same organizations that are in charge of the governance of the blockchain, or again can be implemented just by a single server if there is sufficient trust in getting reliable answers to the binary search queries. Our design is flexible.

In addition to K, the infected person can also upload the root of a Merkle tree where the leaves contain committed information (e.g., about BLE signal, location, body temperature) that later on the infected person might like to share to epidemiologists. The binding of the commitment is important to avoid that such information are adaptively changed. The hiding through a Merkle tree is important to leave the ownership of this information to the person until she decides to selectively disclose it.

We remark that avoiding that two smartphones with pseudonyms A and B upload the same K (this would leak some information), is straightforward: A could just upload $H(K|A|B)$ while B could just upload $H(K|B|A)$ where H is a cryptographic hash function.

Shortening pseudonyms. Current standards suggest at least 256 bits for a group element to safely run the DH protocol over elliptic curves. This size however exceeds the space available in a BLE beacon. Moreover we really stand for defeating mass surveillance attacks and therefore we suggest to be more conservative, using 384 or even 512 bits. While one might think to resolve the issue of the small space in a BLE beacon by just resorting to very short (and therefore in our view too risky in case of mass surveillance attacks) keys [a8x20a] we propose instead a different approach: we decouple the group element from the pseudonym.

Recall that following also previous work, a value announced in a BLE beacon should last only for a few minutes, to then be replaced by a new one. The smartphone will periodically generate new independent group elements for DH and will keep them locally. Since they are too large to be sent in BLE beacons, the smartphone will upload them to a bulletin board. Again, our design is flexible and the bulletin board can be maintained by a server or alternatively be implemented with a blockchain. As above, we support the second option since it gives full decentralization and makes more citizens willing to participate, having more chances to defeat the virus.

Our choice of decoupling the group element from the pseudonym is implemented by setting the 128 bit[5] pseudonym as the address on the bulletin board of the corresponding group element. By using as pseudonym a short representation of the group element, we need a different mechanism to implement the key exchange. Recall that the infected person must compute the key K and push it to the server, while the non-infected person needs to compute the key K to then pull (if it exists) K from the server[6]. Starting from a short pseudonym every player will recover the actual group element from the bulletin board that records all group members. This is a fast operation since the pseudonym is the address of the group element and thus there is no need to download a large amount of data or to do any expensive search.

[5] This is the size for a pseudonym that is commonly allowed by BLE beacons.

[6] Recall that we suggest to perform binary search to speed up this phase and to avoid the download of excessive amount of data.

Silent tracing. Our system being based on virtual anonymous call for whoever has been in close proximity with a recently detected infected person, is immediately secure with respect to silent tracing. Indeed when a person walks alone and passes by a silent tracing device, the sole transmission of the pseudonym used in that moment by the smartphone does not allow to understand if later on that person is infected, since there will be no key K that can be found.

Shameless tracing. A government can also try to trace citizens by having on its territory many devices that behave as smartphones, therefore announcing pseudonyms with the hope of receiving a call or making calls in order to infer some information on the locations and identities of the citizens. It goes without saying that this is a very easy to detect attempt. Indeed the smartphone application could easily inform the owner at any time on the number of BLE beacons that are currently received. Therefore citizens can realize the existence of a malicious device and ask police to destroy them. Notice that the only dangerous BLE devices are the ones that announce the very specific beacon for the location tracing system. There are specific codes to differentiate beacons for different systems. Therefore, in our system, it is still completely fine (i.e., they do not have to be destroyed) to have on the territory devices (e.g., information kiosks) that use BLE to provide other services.

Unlinkability over TCP/IP, timing, and other side-channel attacks. As in all ACT systems, the person owning a smartphone could be identified through the IP address when connecting to servers. Moreover when uploading a batch of group elements some attention should be paid so that they are not linkable. We therefore suggest the use of mixnets, programmed delays, onion routing and periodic uploads of bogus data with the only purpose to confuse and make harder to achieve any profiling attempt.

Replacing DH with other key-exchange protocols. We have proposed the DH protocol because it is computationally efficient and has very low space requirements. Nevertheless our design is flexible and one can use other key-exchange systems as long as there is just one message per party that is moreover independently computed from the other message.

Countermeasures to DoS attacks. Typical DoS attacks can be mitigated with pretty standard approaches, just to mention some: CAPTCHAs, proofs of work, anonymous tokens.

Removing old data from the bulletin board (even from the blockchains). The entire information available on the bulletin boards does not disclose identities. Moreover it does not allow to link keys with pseudonyms. Nevertheless, in order not to overload servers with old information (e.g., anything uploaded more than 20 days ago), past data can be removed from the bulletin board pretty easily. If the bulletin board is managed by a server, then old data can just be deleted. If instead the bulletin board is realized through a blockchain, then we suggest that periodically the pointer to the genesis block moves forward to the next block. Essentially the blockchain will always consists of the blocks generated in the last relevant time period (e.g., 20 days). Moreover this process can be made even more transparent by uploading every 10 minutes on the Bitcoin blockchain the cryptographic hashes of the blocks generated in the last 10 minutes. This allows everyone to constantly verify that the bulletin board is correctly decentralized and redacted.

Remark on the actual realization of the Pronto-C2 system. As far as we understand, the main obstacle to a realization of the Pronto-C2 system is the decision of Apple&Google to have pseudonyms chosen according to their design only. We hope that this will change soon and Apple&Google will allow also other systems like ours to have the possibility of rotating the MAC address of the BLE device synchronously with the rotation of the pseudonyms.

2 Brief Description of DP-3T System

In this section, we briefly describe the functioning of the DP-3T system as reported in the white paper [DT20a]. Two versions of the system are described: the first one, termed as "low-cost", is more efficient but provides lower privacy guarantees than the second one, which is termed "unlinkable".

Low-cost design. As every contact-tracing system following the principles of decentralization, smartphones broadcast locally generated ephemeral pseudonyms (EphIDs) via BLE advertisements.

Whenever a smartphone detects an incoming EphID, it registers this pseudonym EphID along with a coarse time information and every data which might be needed later to compute the risk of contagion (e.g., signal strength, duration of the contact). As the word ephemeral suggests, the broadcasted pseudonyms are periodically changed to prevent tracing.

All the EphIDs that a device will ever generate can be deterministically derived from a short uniformly random secret-key sk_0. At each day t, a new secret-key is derived as $\mathsf{sk}_t = H(\mathsf{sk}_{t-1})$ where H is a cryptographic hash function.

Starting from sk_t the whole set of the EphIDs for the day t, is determined partitioning in 16-byte chunks a string whose length depends on how frequently the EphIDs are changed. Such string is computed as $\mathsf{PRG}(\mathsf{PRF}(\mathsf{sk}_t, c))$ where PRF is a pseudo-random function, c is a fixed public string, and PRG is a stream cipher.

When a user is tested positive, she uploads the pair (sk_t, t) to a back-end server which is trusted to provide this information to all the other users and to check that the uploads are performed by authorized users therefore preventing the dissemination of false positives. This authorization check is not explained in details in [DT20a]. After this step, the infected user's device disappears by the application scenario and her device generates a completely new random secret-key sk_0.

Each user can periodically query (e.g., at the end of the day) the backend server in order to get an update on the new pairs that have been added to the system. Given these pairs, the device can generate the corresponding values EphIDs seeking for matches in its local contact database. If a match is found, the risk of infection is computed given the auxiliary information and the user is notified when needed.

Unlinkable design. In order to get better privacy guarantees at the cost of a larger volume of downloads needed by the smartphone, in their white paper the DP-3T's team proposes a slightly different design which they term unlinkable.

In this design, different EphIDs are randomly and independently generated in the following manner: when a new ephemeral pseudonym is needed, the smartphone generates the ephemeral pseudonym EphID_i as $\mathsf{TRUNCATE}_{128}(H(\mathsf{seed}_i))$.

Smartphones store all the seeds used in a relevant time window (e.g., 14 days). When a patient is tested positive, she can selectively decide which pseudonyms she wants to communicate to the server (e.g., she can exclude pseudonyms emitted when she met only one person).

After this decision has been made, the smartphone uploads the set composed by the selected pairs $(seed_i, i)$. Upon receiving them, for each pair the server computes $H(\mathsf{TRUNCATE}_{128}(H(\mathsf{seed}_i))||i)$ and inserts it in a Cuckoo filter [7]. Such filters are generated and made available to the users on a regular basis.

Each smartphone uses these filters to determine if contacts with infected individuals occurred. In this regard, the smartphone checks the inclusion of all its recorded ephemeral pseudonyms into the filters.

3 Mass Surveillance

Mass surveillance is an activity put in place to watch, even discontinuously, over a substantial fraction of the population by monitoring, for example, their movements and/or habits.

Even though decentralized solutions guarantee, in general, better privacy compared to centralized ones, mass surveillance is still a possible threat and must be mitigated as much as possible when introducing new intrusive technologies.

Unfortunately, the DP-3T's low-cost design, as acknowledged by the DP-3T team (cf. SR4 in [DT20b]), opens up the mass surveillance of infected users over the contagion time window. Since it is fairly possible than soon or later everyone will be infected, this means that a very large percentage of the population could be controlled, at least for a time window. This mass surveillance attack can be performed even by an attacker not colluding with the server or the health authorities.

In particular, an attacker can locally store all the observed pseudonyms along with a fine-grained time and location log. Since all the EphIDs of a user are deterministically defined by the announced secret-key, the attacker is able to link pseudonyms that belong to the same infected individual and can, therefore, leverage this information to track a user's path over the contagion period. The tracing is limited to the contagion time window and is relative only to infected individuals. Although the impact of this attack could seem limited at a first glance, it can easily scale up to way more creepy scenarios.

In the following paragraphs, we present several possible attacks towards contact tracing systems which, when successful, undermine users' privacy, eventually leading to undetectable mass surveillance attacks. Furthermore, we evaluate and compare the resilience of DP-3T and our Pronto-C2 system (see Section 4) against such attacks.

Our attacks are inspired by the work of Vaudenay [Vau20] and by the issues reported in the DP-3T git repository [a8x20b, a8x20a]. We carefully take into account these issues and attacks to illustrate more precise scenarios unveiling significant mass surveillance attacks.

3.1 ATK 1: Tracing of Infected Users With Trusted Server

- **Attacker's capabilities**: Anyone with enough economical resources. The attacker has the ability to install, in a sufficiently large number of different locations, passive BLE devices. The only capability of a passive device is to operate over BLE channels in reception mode. We also assume that such devices are provided with enough memory to store a significant amount of received data (i.e., pseudonyms and auxiliary information).

[7] A Cuckoo filter is a space-efficient probabilistic data structure used to test whether an element is a member of a set. False positive matches are possible, but false negatives are not.

- **Attack description**: The passive devices record the observed pseudonyms along with a fine-grained time log. The location of each device is fixed and determined by the attacker. When a user is tested positive and uploads data into the contact tracing system, the system itself provides related data to all the users. The attacker then combines this data with his logs.
- **Attack's outcome**: The attacker computes a fine-grained tracing of infected users during the contagion time window. Furthermore, the attack is practically undetectable by the users since the attacking devices operate only in reception mode.

The low-cost design of DP-3T is vulnerable to ATK 1. It is not difficult to imagine the feasibility of such an attack, as an example one could consider a company with many stores spread over the territory. This corporation can have an interest in tracking infected costumers, even if it is not particularly interested on their health conditions, in order to use their movements to perform accurate profiling without costumers' consent.

What is needed is merely the capability to install, in a sufficiently large number of different locations, passive BLE devices recording the received EphIDs. The attack is carried out as described in Section 2. In the scenario we envision the amount of gathered data can be, given a relatively small effort, very large, thus resulting in a possibly very fine-grained tracing.

The key issue of the low-cost design, leading to the applicability of ATK 1 lies in the fact that when the secret-key of an infected person is added to the system everyone can derive all the related EphIDs, enabling the linking of pseudonyms with infected individuals. We point out that this attack is practically undetectable, at least at the application level, since the devices do not need to propagate any signal. Given the huge impact that this easy-to-deploy attack can have on users' privacy, the low-cost design appears utterly unsuitable for practical deployment, unless one wants to give up on protecting citizens from mass surveillance attacks.

3.2 ATK 2: Tracing of Infected Users With Colluding Server

- **Attacker's capabilities**: The attacker is the same as ATK 1. However, in addition, he can collude with the server. Note that the server could be under a significant influence of the government.
- **Attack description**: The attack is analogous to the one described in ATK 1. The only difference is that, along with data provided to all regular users, the attacker receives all data that are in possession of the server.
- **Attack's outcome**: see ATK 1.

Unlinkable design of DP-3T is vulnerable to ATK 2. Since the Cuckoo filter allows users to know only the infect's EphIDs they previously observed, this design succeeds in preventing ATK 1. However, the claim that "infected people in the unlinkable design are not traceable", as affirmed in [DT20a] is oversimplified and requires a deeper treatment. In fact, such claim is true only with respect to attackers who do not cooperate with the server. Considering also the fact that governments might have control over the servers, an attack similar to the one described for the low-cost design can be put in place.

The devices listening on the BLE channels could be deployed or hidden in many ways. An example are smart kiosks, which are already used in many cities to provide useful functionalities to the citizens. For the purpose of the description, we will refer to all possible passive devices as kiosks. The attack works as follow:

1. Each kiosk collects the information of people that pass near the device, the information stored is composed of EphID_i and a fine-grained time log.

2. At the end of the day, each kiosk downloads the filters F from the server.

3. Each kiosk checks if the collected EphIDs are included in the filters.

4. The attacker, that controls the kiosks and colludes with the server, obtains from the server all the seeds of the infected citizens.

5. The attacker matches the EphIDs of his records with the ones generated form the seeds of the infected individuals, thus tracing the infected individuals who passed nearby the kiosks over their contagion time window.

The element of centralization in DP-3T requiring the server to compute the Cuckoo filter of the EphIDs, enables mass surveillance with low overhead. Moreover, it is almost impossible to determine if a process of surveillance is actually active or not.

Another important point is that governments can do a further step associating a pseudonym to the real identity of an infected user: whenever there is a police checkpoint to control people, the police can be instructed to collect EphIDs and associate them to the name and surname of the controlled persons. When a person is tested positive, the government can check data collected by the police. If one of the EphIDs comes from the seed of an infected person, the governments can obtain all the movements of this citizen during the contagion time window.

The same thing can happen when a citizen gets tested for SARS-CoV-2. In fact the tests are typically performed after some form of identification. If the citizen's smartphone application is active while she is in the laboratory, her EphIDs can be detected by a kiosk. If the citizen is eventually tested positive and uploads the seeds related to her visit to the lab, a match between her real identity and movements during the contagion time window can be easily exposed.

Remark I. It is worth noting that those tracking strategies are not only theoretical speculations. Let us consider unpopular citizens or political dissidents, like anarchists. In many countries these persons are observed by governments who want to track them and discover their contacts. Let us say that A is a dissident and consider the following possible scenarios:

- If A goes to a medical laboratory to check if he is infected, the government can force the laboratory to communicate to A that he is positive to SARS-CoV-2. At this point A can choose to send the seeds used in the contagion time window to the server who puts them in the Cuckoo filter. If A sends these data to the server, the Big Brother can track all his movements in the last days.

- After A is declared positive to SARS-CoV-2 and he sends his seeds to the server, it is very likely that someone among his contacts would desire to perform the medical test as well. This opens the possibility for the government to track all these persons and attack their privacy possibly linking them to the dissident.

- In another possible scenario the government could control whether a close relative P of A gets tested and force the laboratory to notify P the positivity to SARS-CoV-2. In this case it is likely that A will go to the medical laboratory to get tested for the virus as well. From now on, this scenario is analogous to the previous ones.

Finally, we want to point out that, even if the unlinkable design solves in part the issue of linkability of the pseudonyms, the server gets more power because it can add every EphIDs that it gets to know in the Cuckoo filter. This can cause additional false positives.

Remark II. The idea of having kiosks spread over the territory could seem somewhat artificial. However, as stated by PACT [Tea20], it is possible to justify kiosks as a way to add functionalities to contact tracing systems. In particular, the authors state that there should be a way to inform persons if a surface can be infected due to the proximity of an infected person. Therefore, in their system, kiosks actively participate to the protocol registering and relaying pseudonyms of people who have been in proximity of the kiosks. By doing so, the kiosks could inform people about the risk of having been in contact with a contaminated surface. In this system, where kiosks are active players and are justified to actively propagate BLE messages, there would be no ways to distinguish malicious kiosks from honest ones. In turn, this could easily open doors to mass surveillance programs operated by governments without being detected.

3.3 ATK 3: Leakage of the Contacts of the Infected Users

- **Attacker's capabilities**: The attacker has the same capabilities of a regular user.
- **Attack description**: When users are tested positive, they upload data to the system. The attacker uses such data to compute additional information beyond his own risk factor.
- **Attack's outcome**: The attacker can compute data on contacts of infected users such as the number of their contacts and co-location with other infected users.

Systems in which the infected users upload an encoding of the observed pseudonyms are more prone to this attack since the content and the amount of communicated data depend on the actual number of experienced contacts. One could think to mitigate this issue by putting a bound on the number of contacts that a user can notify. However, it is not evident what is the appropriate value for this bound to effectively fight the pandemic. Also, co-location of infected users is more likely to be exposed since infected users who met each other might end up reporting some linkable encodings. If at some point two infected users met each other, the information that these users send to the server can enable the reconstruction of clusters of infected users who have been co-located. Nevertheless, it is extremely hard to imagine how such leakage could be exploited by mass surveillance attacks that are the focus of our work. We remark that systems like DP-3T are not affected by this attack.

3.4 ATK 4: Creation of Mappings Between Real Identities and Pseudonyms

- **Attacker's capabilities**: The attacker is composed of the server and the health authority.
- **Attack description**: The attacker exploits the authorization mechanism, also used to avoid uploads of false positives, to find a mapping between the real identity of a user and her pseudonyms.

- **Attack's outcome**: a mapping between the real identity of a user and her pseudonyms.

Every scheme where the authorization mechanism consists of simply forwarding to the server data (i.e., an authorization code) provided by the laboratory, is vulnerable to this attack. In fact, the laboratory can communicate the mapping between the authorization code and the real identity to the server, which can in turn derive the mapping between this code and data uploaded by the user (i.e., pseudonyms). The authorization mechanism is not made explicit in many relevant proposals, as well as in DP-3T. A reason advocated for this choice is flexibility to different deployment scenarios. However, we want to point out that the way this check is performed reflects into serious implications on users' privacy.

3.5 ATK 5: Proving Contact With an Infected User

- **Attacker's capabilities**: The attacker has the same power of a regular user of the system. Additionally, he might get access to a service making him able to prove the ownership of some data in a specific point in time (e.g., with the help of a blockchain).
- **Attack's outcome**: The attacker provides a convincing proof of him having been in contact with an infected user.

However, while on one hand the ability to successively conduct this attack can lead to some threats, on the other side it could be seen as a feature.

- Attack: An evil guy A could promise cash in exchange of proof of co-location with an infected user. If different users provide this proof to A in exchange for his money, A could then build co-location information between a larger group of users.
- Feature: Suppose that, unfortunately, at a certain point during the pandemic, laboratories are flooded of people asking for tests. In this scenario, having a way to prioritize such requests could be of great utility. In fact, malicious users may try to fake risk notifications only because they want to get tested even if it is not needed. Therefore, being able to provide a convincing proof of contact could help honest users to get tested quickly.

DP-3T is vulnerable to ATK 5. The attack is really straightforward. Whenever the attacker receives a pseudonym she commits it to a blockchain, together with an information related to her real identity. When she wants to prove that she was in possession of this pseudonym at a specific point in time, she opens the commitment.

In the low-cost design, the infected user's secret key also determines the time-slot in which the pseudonym was transmitted, and that can be matched with the temporal information of the blockchain (of course confirmation times must be taken into account) in order to be assured, at a certain extent, that the pseudonym was indeed recorded by the attacker's device via BLE rather then received via other means. In the unlinkable design, the proof is limited in demonstrating knowledge of the infected user's pseudonym prior to the publication of the filter.

Viewing ATK 5 as a feature to help epidemiologists is problematic in DP-3T. The DP-3T white paper [DT20a] proposes that, only if they are willing give up some of their privacy, users can share additional data with epidemiologists to help them in their analysis. However, the way this functionality is implemented, at least as in the current version of DP-3T white paper [DT20a], presents some shortcomings in both designs.

In the low-cost design, when users are notified a possible risk, they can optionally anonymously upload, along with useful metadata, the secrets associated to the infected individuals they met. In the unlinkable design, instead, the user provides a list containing the number of infected users' pseudonyms observed per-day during the contagion time window.

However, in both cases, there is no way whatsoever to verify the legitimacy of data. In this regard, what might be needed is a way to verify that the contact actually took place leveraging ATK 5 as a feature. However, even though it is possible, it seems, at least at a first glance, that providing proof of having had a contact with an infected person is not easily scalable to a considerable portion of the users. Furthermore, there would still be the need in trusting the correctness of the metadata provided by users.

4 Pronto-C2: Design and Analysis

One of the main drawbacks in previous solutions, in particular in DP-3T [DT20a] and PACT [Tea20] systems (in all their variants) is the possibility for an attacker to test weather a set of pseudonyms belongs to an infected person and thus to infer the victim's movements. The problem is evident in the basic DP-3T protocol but, as analyzed in Section 2, also arises in the DP-3T's "unlinkable" variant.

Our approach diverges radically from DP-3T in that we turn the paradigm upside down. In our system it is the infected to be responsible to publish the data of the people with whom he/she got in touch. It is up to each participant to verify the occurrence of a risk. This is done through careful use of cryptography, without sacrificing performances.

Pronto-C2 in a nutshell. In a nutshell, Pronto-C2 works as follows. We assume the generator g of an elliptic curve group of prime order is known to all participants. For simplicity we will describe our scheme using a server Server that manages a bulletin board accessible to all participants. As explained in Section 1, our design is flexible, we can have blockchains or just servers depending on the desired level of transparency and performance.

Periodically, each user U performs the following update operation. Let i be the current time slot. U setups a set of ephemeral- and secret- keys $(\mathsf{Eph}_{\mathsf{U},i+j} = g^{\mathsf{sk}_{\mathsf{U},i+j}}, \mathsf{sk}_{\mathsf{U},i+j}), j = 0, \ldots, n-1$ for some parameter n. For $k = i, \ldots, i+n-1$, U sends to Server the string $\mathsf{Eph}_{\mathsf{U},k}$ and privately stores the address $\mathsf{addr}_{\mathsf{U},k}$ in which $\mathsf{Eph}_{\mathsf{U},k}$ appears on the bulletin board. The idea is that these addresses will be used for the next n time slots. Each n time slots U runs again the update operation, previous keys are not overridden.

At each time slot i, user U proceeds as follows. U broadcasts addr_i and listens for addresses sent by other users. Each address received can be recorded along with auxiliary information.

Consider a simple scenario in which Bob got a diagnosis for COVID-19 and he has been in close proximity to her neighbor Alice at time i (among many other contacts). Let us denote by $\mathsf{Eph}_A = g^{\mathsf{sk}_A}$ (resp., $\mathsf{Eph}_B = g^{\mathsf{sk}_B}$) Alice's (resp., Bob's) ephemeral-key at the time of the contact. Bob computes the "shared key" $K = \mathsf{Eph}_A^{\mathsf{sk}_B}$ and uploads it to Server after requiring the laboratory to blind sign it. We require signatures by the laboratories to prevent DoS attacks and we use blind signatures to prevent the laboratories to link patients to information on the server.

We assume that Server accepts only keys with valid signatures. Moreover, the server can collect them in an *ordered* set, to allow users to search for a specific key in logarithmic time. Here the assumption is that smartphones do not want to download large amounts of data but however the validity of the answers can be massively detected by computers that continuously make sure that the server is working properly. Moreover if a server misbehaves, there is a clear evidence of cheating against the reputation of the server's managers.

At the end of the day, if Alice wants to know whether she has been in contact with an infected person, she does the following. For each address she received from a nearby user, she retrieves from Server the corresponding ephemeral-key so she has the Bob's ephemeral-key Eph_B. She computes $K = \mathsf{Eph}_B^{\mathsf{sk}_A}$, search on Server for occurrences of K and if K is present she is notified the risk.

Pronto-C2's system and crypto ingredients. The ingredients of our system are:

- A secure elliptic curve group of prime order p. We assume a generator g of the group to be publicly known to all participants.
- A blind signature scheme. The blind signature is used only to authorize a specific laboratory to sign user's data while hiding the message. We defer to [Cha83, Cha88, PS96] for the syntax and security properties of blind signatures.
- A server Server that is used as a bulletin board (see previous discussion and Section 1). The server allows any user to write data of the type "ephemeral-keys" whereas, in order to write a data of the type "key", a valid (blind) signature issued by one of the authorized health authorities has to be provided. Keys will be written on the server only if the signature is valid.
- We assume the smartphone application to communicate with Server via TOR [TOR] and in particular ephemeral-keys are communicated by means of both mix-nets [Cha81] and TOR [TOR]. TOR is used to break the link between ephemeral-keys and real identities and mix-nets make difficult to figure out whether two ephemeral-keys belong to the same user. Alternative solutions are also possible, but they depend on the specific context in which the system operates, therefore for now we remain generic. See also the discussion in Section 1.

Pronto-C2's setting and actors. The actors involved in our protocols are:

- The users who run a smartphone application endowed with a BLE beacon. A generic user will be denoted by U.
- A set of health authorities (HAs) who can engage with users in medical examinations and tests for the virus.
- The server (Server) that manages the bulletin board.

The Pronto-C2 system. The Pronto-C2 system is described by the following phases and events. During the execution of the system, each user U keeps a set P_{U} that is empty at the onset. We assume each user U to keep an internal variable called *time slot*. At the start of the protocol U's time slot is set to 0 and each X seconds the time slot is increased by 1. X is a parameter of the protocol (e.g., 300 seconds).

In the Setup Phase each participant runs as follows.

- U: configure the smartphone application and set the time slot to 1.
- HA: setup the parameters for the blind signature scheme and publish the public-key.
- Server: perform any necessary step to accept incoming read and write requests. Publish the public-keys of the authorized HAs.

Figure 1: Setup Phase.

In the Update Phase executed at time slot i, each user U interacts with Server as follows.

- $\mathsf{U} \rightarrow \mathsf{Server}$: for $j = 0, \ldots, n-1$ generate a pair of ephemeral- and secret- keys $(\mathsf{Eph}_{\mathsf{U},i+j} = g^{\mathsf{sk}_{\mathsf{U},i+j}}, \mathsf{sk}_{\mathsf{U},i+j})$ drawing an element $\mathsf{sk}_{\mathsf{U},i+j}$ at random from $\mathcal{Z}_p$.[a] For $j = 0, \ldots, n-1$ upload $\mathsf{Eph}_{\mathsf{U},i+j}$ to Server and store the address addr_{i+j} in which $\mathsf{Eph}_{U,i+j}$ appears on the bulletin board.

HAs do not perform any operation.

[a] To optimize the space, the user could choose a single seed s during the Setup Phase and in each time slot i derive $\mathsf{sk}_{U,i} = \mathsf{PRF}(k, i)$.

Figure 2: Update Phase.

- Setup Phase. There is a setup phase in which all the involved actors perform the basic setup described in Figure 1.
- Update Phase. There is an Update Phase, described in Figure 2, that is run periodically by each user U each n time slots (i.e., when U is at time slot j and j is a multiple of n).

  We assume each time slot to be short enough to prevent significant linkage of ephemeral-keys to users' movements, but long enough to correctly evaluate exposure risks. Moreover, we assume the parameter n to be sufficiently large to not require the users to perform the expensive Update Phase too frequently (e.g., n can be set so that the update is performed each week).
- Broadcast Phase. There is a Broadcast Phase, described in Figure 3. The Broadcast Phase is run multiple times within the time slot. The frequency with which this phase is executed within a single time slot is another parameter of the protocol.
- Listen Event. The Listen Event, described in Figure 4, is triggered when a BLE message is received.
- Test Positive Event. The Test Positive Event is triggered when a user tests positive for SARS-CoV-2 at one of the laboratories of one of the HAs. When a user U gets a positive result

In the Broadcast Phase, each user U proceeds as follows.

- U: Let i be the current U's time slot. Broadcast the address addr_i generated in the last Update Phase using BLE.

Other participants (HAs and Server) do not perform any operation.

Figure 3: Broadcast Phase.

When a BLE message is received, the Listen Event is triggered and each user U proceeds as follows.

- U: let addr_R be the address contained in the received message, i the current time slot and t any other auxiliary information (e.g., BLE signal, location, body temperature, time).

  Add $(\mathsf{Eph}_{\mathsf{U},i}, \mathsf{sk}_{\mathsf{U},i}, \mathsf{addr}_R, t)$ to the set P_{U}, where $\mathsf{Eph}_{\mathsf{U},i}$ (resp., $\mathsf{sk}_{\mathsf{U},i}$) is the ephemeral-key (resp., secret-key) that U computed in the last Update Phase.

Other participants (HAs and Server) do not perform any operation.

Figure 4: Listen Event.

for SARS-CoV-2 at HA's lab, U chooses a subset P'_{U} of P_{U}. U can decide upon which time slots to insert in P'_{U} based on any arbitrary criteria (e.g., can exclude time slots in which U suspects to have met some people to whom he wants to hide his disease). When the event is triggered, U interacts with Server and HA as depicted in Figure 5.

- Verify Phase. The Verify Phase, described in Figure 6, is carried out by a user U who wants to discover whether she got in contact with some other user U^+ who got a positive result for SARS-CoV-2.

4.1 Analysis of Pronto-C2

In this section we show that Pronto-C2 withstands all the attacks shown in Section 3, and moreover ATK 5 can be considered a feature to help epidemiologists.

- ATK 1: Tracing of infected users with trusted authorities (cf. Section 3.1).
  Pronto-C2 withstands this attack since what an infected user uploads to the server is a DH key that cannot be linked to a specific ephemeral-key without the corresponding secret-key or secret-keys of people with whom the user has been in contact.

- ATK 2: Tracing of infected users with colluding authorities (cf. Section 3.2).
  Pronto-C2 is secure against this attack since the server and the authority cannot link their information to track the infected user. The key stored on the bulletin board cannot be "decrypted" by the authorities since the authorities do not possess any key exchanged by

- $\mathsf{U} \leftarrow \mathsf{Server}$: choose a subset P'_{U} of P_{U} and for each quadruple $(\mathsf{Eph}_{\mathsf{U}}, \mathsf{sk}_{\mathsf{U}}, \mathsf{addr}_R, t) \in P'_{\mathsf{U}}$, retrieve from Server the ephemeral-key Eph_R stored at address addr_R. Compute $K = \mathsf{Eph}_R^{\mathsf{sk}_R}$.
    - ∗ Interaction between U and HA: for each value K computed by U as before, U and HA interact to compute a blind signature σ of K.
    - ∗ $\mathsf{U} \rightarrow \mathsf{Server}$: for each K computed by U as before, upload K and σ to Server.
    - ∗ $\mathsf{Server} \leftarrow \mathsf{U}$: upon receiving any pair (K, σ) from U, verify σ and if the signature is valid add K to the efficient data structure.

Figure 5: Test Positive Event.

When a user U wants to verify whether she got in contact with any user U^+ who got a positive result for SARS-CoV-2, U engages in an interactive protocol with Server as follows.

- $\mathsf{U} \leftarrow \mathsf{Server}$: Let P_{U} the set computed by U during the protocol execution so far. For each quadruple $(\mathsf{Eph}_{\mathsf{U}}, \mathsf{sk}_{\mathsf{U}}, \mathsf{addr}_R, t)$ in P_{U} do the following:
    - ∗ Retrieve from Server the ephemeral-key Eph_R located at address addr_R. Compute $K = \mathsf{Eph}_R^{\mathsf{sk}_{\mathsf{U}}}$ and search for K on the Server.[a] If K is present, compute the risk and notify the user.

HA does not perform any operation.

[a] As we described the protocol, the user does not directly check the signature since it is the smart contract to ensure that only keys with valid signatures have been uploaded to Server. For a stronger verifiability guarantee we can change the protocol so that the user is given the possibility to download and check the signature.

Figure 6: Verify Phase.

users. Here is crucial that the kiosks are passive, otherwise their interaction with infected users will cause their traceability.

- ATK 3: Leakage of the Contacts of the Infected Users (cf. Section 3.3).
  Pronto-C2 is not immune to this attack due to the structure of the protocol: indeed all data belonging to a single infect user U can be stored on the bulletin board by only providing a single blind signature produced by some HA. This means that it is possible to learn how many users U encountered during the infection window and the laboratory in which the test was done, a reasonably small leakage. As reported in Section 3.3, along with a mitigation strategy, Pronto-C2 could also leak some co-location information about infected individuals. There is a solution to make Pronto-C2 secure against the leakage of some co-location information. We modify Pronto-C2 as follows: A (resp. B) notifies the risk to B (resp. A) by uploading to the server the key $K_A = H(K||A||B)$ (resp. $K_B = H(K||B||A)$). In this way, the two keys are unlinkable.

  It is important to point out that the unlinkability of the keys introduced by uploading $K_A = H(K||A||B)$ instead of K contradicts the idea that DP-3T's risk analysis (SR 6) [DT20b] seems to suggest when claiming that "For epochs in which groups of at least three people were in close proximity to each other, this will reveal temporal co-location information about infected individuals to the server".

- ATK 4: Creation of mappings between people and pseudonyms (cf. Section 3.4).
  Pronto-C2 withstands this attack since the authorization code provided by HA is a blind signature of the ciphertext produced by the infected user U. This guarantees that HA cannot link the real identity of U to the records he previously stored in the bulletin board.

- ATK 5: Proving contact with an infected user (cf. Section 3.5).
  Pronto-C2 is subject to this attack, but, on the other side, Pronto-C2 guarantees that every time a user U claims to have been in proximity of an infected user, U has also the ability prove it. However it must be noted that if U decides to reveal her secret-key, she is exposed to silent tracing for the duration of one time slot.

Viewing ATK 5 as a feature to help epidemiologists. As stated in Section 3.5, being able to prove the contact with other infected users might be useful, especially to provide data to epidemiologists who are studying the virus. In Pronto-C2, doing so is pretty straightforward since, in order to prove the contact, it suffices for a user to provide the ephemeral- and secret- keys related to the corresponding record on the bulletin board.

It would be also useful to add other epidemiological information while preserving privacy. This is easily done by posting on the server a succinct commitment (e.g., the root of a Merkle tree) to the auxiliary information that the infected user collected. (Such commitment has to be blindly signed by a HA as for the key K.) The succinct commitment has the property that the user who created the commitment can securely and selectively disclose partial information about the committed data to any participant of his choice. Furthermore, the hiding property can prevent distinguishing users who do not want to share any form of meta-data at all. In fact such users can provide a random string instead.

Attacks	**Description**	**Low-cost DP-3T**	**Unlinkable DP-3T**	**Pronto-C2**
ATK 1	Tracing of infected users with trusted server	✗	✓	✓
ATK 2	Tracing of infected users with colluding server	✗	✗	✓
ATK 3	Leakage of the contacts of the infected users	✓	✓	✗
ATK 4	Creation of mappings between real identities and pseudonyms	✗	✗	✓
ATK 5	Proving contact with an infected user	✗	✗	✗

Figure 7: Identified attacks. We show which system is susceptible to which attack. ✗ denotes a system which is vulnerable to the attack, ✓ a system which is safe against an attack, finally ✗ denotes an attack with minimal leakage (cf. Section 4.1).

Security enhancement. In Pronto-C2 if A notifies a risk to B and B notifies a risk to A, they will both upload to the server *identical* keys and in some circumstances this may be a leakage of non-trivial information. The issue is easily solved: A (resp. B) notifies the risk to B (resp. A) by uploading to the server the key $K_A = H(K||A||B)$ (resp. $K_B = H(K||B||A)$). In this way, the two keys are unlinkable.

5 Conclusion

An unprecedented social pressure is pushing towards the adoption of contact tracing system in response to the COVID-19 pandemic. Any tracing application introduces new dangers to users. The proven existence of previous attempts to realize mass surveillance programs unveiled by Snowden should urge for a deep and careful scrutiny of the emerging solutions that claim to achieve *privacy-preserving* proximity tracing.

In particular, we have analyzed the DP-3T system that has been endorsed by Apple and Google. Currently the DP-3T teams and Apple&Google are working together for making possible the deploy of the DP-3T system. Our analysis shows that there are risks that such system can be abused by governments interested in realizing mass surveillance programs. While one can be happy about giving up privacy in order to obtain a more effective response to the spread of the virus, we insist on the fact that the most privacy preserving solution should be used among the ones that are equally useful for epidemiologists, as advocated in [doc20].

We have then shown our new system named Pronto-C2 that is arguably better in defeating mass surveillance attacks and is at least as good as DP-3T in providing data to epidemiologists. In Figure 7 we compare Pronto-C2 with DP-3T in relation to the mass surveillance attacks described in Section 3.

Acknowledgments. We thank Luigi Di Biasi and Stefano Piotto and for useful discussions on their system[8] as well as their comments on decentralized vs centralized approaches in ACT. We also thank Aggelos Kiayias for suggesting in general the possibility of using blockchains in ACT systems and Alessandra Scafuro for constructive comments on our work. This research has been supported in part by the European Union's Horizon 2020 research and innovation programme under grant agreement No 780477 (project PRIViLEDGE) and in part by the region Campania.

References

[a8x] a8x9. a8x9. `https://github.com/a8x9`. 1.2

[a8x20a] a8x9. DP-3T. `https://github.com/DP-3T/documents/issues/66`, 2020. 1.2, 3

[a8x20b] a8x9. DP-3T. `https://github.com/DP-3T/documents/issues/210`, 2020. 3

[ABB+18] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolic, Sharon Weed Cocco, and Jason Yellick. Hyperledger fabric: a distributed operating system for permissioned blockchains. In Rui Oliveira, Pascal Felber, and Y. Charlie Hu, editors, *Proceedings of the Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal, April 23-26, 2018*, pages 30:1–30:15. ACM, 2018. 1.2

[AG20] Apple and Google. Apple and Google's exposure notification system. *https://www.apple.com/covid19/contacttracing*, 2020. 1

[BLS19] Johannes K Becker, David Li, and David Starobinski. Tracking anonymized bluetooth devices. *Proceedings on Privacy Enhancing Technologies*, 2019(3):50–65, 2019. 1

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981. 4

[Cha83] David Chaum. Blind signature system. In David Chaum, editor, *CRYPTO'83*, page 153. Plenum Press, New York, USA, 1983. 1.2, 4

[Cha88] David Chaum. Blind signature systems. U.S. Patent #4,759,063, July 1988. 4

[CHRT20] Andrew Clement, Jilian Harkness, George Rain, and Laura Tribe. Snowden surveillance archive. *https://snowdenarchive.cjfe.org/greenstone/cgi-bin/library.cgi*, 2020. 1

[Com20] European Commission. Commission recommendation of 8.4.2020 on a common Union toolbox for the use of technology and data to combat and exit from the COVID-19 crisis, in particular concerning mobile applications and the use of anonymised mobility data. *https://ec.europa.eu/info/sites/info/files/recommendation_on_apps_for_contact_tracing_4.pdf*, 2020. 1

[8] SM COVID-19 App: https://www.softmining.it/index.php/sm-covid19-app/.

[D'A83] Nino D'Angelo. Pronto si tu. https://www.youtube.com/watch?v=8DP3UyDS0Ts, 1983. 4

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976. 1.2

[doc20] Joint Statement on Contact Tracing: Date 19th april 2020. https://drive.google.com/file/d/1OQg2dxPu-x-RZzETlpV3lFa259Nrpk1J/view, 2020. 1.1, 5

[ds] Ministero della salute. Contact tracing: Arcuri firma ordinanza per app italiana. http://www.salute.gov.it/portale/nuovocoronavirus/dettaglioNotizieNuovoCoronavirus.jsp?lingua=italiano&menu=notizie&p=dalministero&id=4513. Accessed: 2020-04-27. 1

[DT20a] DP-3T's Team. Decentralized privacy-preserving proximity tracing. *https://github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf*, 2020. 1, 2, 2, 3.2, 3.5, 4

[DT20b] DP-3T's Team. Privacy and Security Risk Evaluation of Digital Proximity Tracing Systems: Date 21st april 2020. *https://github.com/DP-3T/documents/blob/master/Security%20analysis/Privacy%20and%20Security%20Attacks%20on%20Digital%20Proximity%20Tracing%20Systems.pdf*, 2020. 3, 4.1

[DT20c] DP-3T's Team. Response to 'Analysis of DP3T: Between Scylla and Charybdis': Date 23rd april 2020. *https://github.com/DP-3T/documents/blob/master/Security%20analysis/Response%20to%20'Analysis%20of%20DP3T'.pdf*, 2020. 1.1

[GL20] Mattew Green and Yehuda Lindell. Privacy & tracking to mitigate pandemics: politics and technological solutions. *https://www.brighttalk.com/webcast/17700/392003/privacy-tracking-to-mitigate-pandemics-politics-and-technological-solutions*, 2020. 1

[Gre19] Andy Greenberg. The clever cryptography behind apple's 'find my' feature. *https://www.wired.com/story/apple-find-my-cryptography-bluetooth/*, 2019. 1

[Lau83] Mariano Laurenti. La Discoteca. https://www.youtube.com/watch?v=t9kwU27FG7U, 1983. 4

[PEP20] PEPP-T's Team. Pan-european privacy-preserving proximity tracing. *https://www.pepp-pt.org/*, 2020. 1

[PS96] David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 252–265. Springer, Heidelberg, November 1996. 4

[Tea20] PACT's Team. Decentralized privacy-preserving proximity tracing. *https://pact.mit.edu/wp-content/uploads/2020/04/The-PACT-protocol-specification-ver-0.1.pdf*, 2020. 1.1, 3.2, 4

[TOR] TOR Wiki. https://trac.torproject.org/projects/tor/wiki. Accessed: 2020-04-27. 4

[Vau20] Serge Vaudenay. Analysis of DP3T. Cryptology ePrint Archive, Report 2020/399, 2020. https://eprint.iacr.org/2020/399. 1.1, 3

[Wik] Wikipedia. Bullrun (decryption program). *https://en.wikipedia.org/wiki/Bullrun_(decryption_program)*. 1