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

Gennaro Avitabile*, Vincenzo Botta†, Vincenzo Iovino‡ and Ivan Visconti§
DIEM, University of Salerno
Italy
Email: *gavitabile@unisa.it, †vbotta@unisa.it, ‡viovino@unisa.it, §visconti@unisa.it

***Abstract*—Automatic contact tracing is currently used in several countries in order to limit the spread of SARS-CoV-2. Many governments decided to develop smartphone apps based on the "Exposure Notifications" designed by Apple and Google according to a decentralized approach previously proposed by the DP-3T team. Decentralization was pushed as a key feature to protect privacy in contrast to centralized approaches that could leverage automatic contact tracing to realize mass-surveillance programs.**

In this work, taking into account the privacy and integrity vulnerabilities of DP-3T systems, we show the design of a decentralized contact tracing system named Pronto-C2 that has better resilience against various attacks. We also discuss the significant overhead of Pronto-C2 when used in real-world scenarios.

I. Introduction

The COVID-19 pandemic is currently affecting daily life of many citizens in the world. People are forced to stay home for several weeks; uncertainty, sadness, economic downturn, unemployment, and restrictions on daily activities generate an impelling desire to join any government effort to stop as soon as possible the spread of the virus.

Following recommendations of epidemiologists [13], governments are proposing the use of smartphone applications to allow automatic contact tracing (ACT) of citizens. This raises the question of whether this digital form of contact tracing can be a subtle weapon for governments to violate the privacy of their citizens contributing to new and more sophisticated mass surveillance programs. The vast majority of ACT systems propose the use Bluetooth Low Energy (BLE) to estimate the distance. Among those, many ACT systems (e.g., DP-3T[1], MIT-PACT, UW-PACT and GAEN, the Apple&Google exposure notification system) follow a decentralized approach and claim to guarantee better privacy properties compared to centralized approaches (e.g., ROBERT or NTK).

[1]We say DP-3T to refer to systems proposed by the DP-3T team.

Workshop on Secure IT Technologies against COVID-19 (CoronaDef) 2021
21 February 2021, Virtual
ISBN 1-891562-72-X
https://dx.doi.org/10.14722/coronadef.2021.23013
www.ndss-symposium.org

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 weaknesses of automatic contact tracing systems. We focus in particular on DP-3T (still our analysis is significant also for many other decentralized ACT systems). Considering attacks proposed in recent literature, we discuss how a government can exploit the use of DP-3T to successfully mount privacy attacks contributing of a mass surveillance program. We show that privacy issues in DP-3T are not inherent in BLE-based contact tracing systems, contradicting a claim appeared in one of the documents published by the DP-3T team.

As main contribution of this work, we propose a different system named Pronto-C2 that, in our view, is more resilient against mass surveillance attacks. 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 ACT system where smartphones anonymously and confidentially talk to each other in the presence of the Big Brother.

A. Related Work

Our work mainly focuses on proposing a new ACT system and evaluating its resilience to various attacks in comparison with DP-3T [11]. Still, the issues we discuss are significant to many other decentralized ACT systems such as MIT-PACT [17], UW-PACT [10] and TCN [22]. We remark that almost all vulnerabilities of DP-3T also apply at least in part to the currently widely used ACT systems based on GAEN [2].

DP-3T (and similar systems) exposes several vulnerabilities. Attacks can be carried out not only by the government, but also by unknown adversaries. These vulnerabilities have been explicitly mentioned in [10] (Section 3.1.3): *"This can be abused for surveillance purposes, but arguably, surveillance itself could be achieved by other methods"*.

Several vulnerabilities of DP-3T have been previously analyzed in various works [23], [24], [18], [21]. Vaudenay [23], [24] presented a detailed list of both privacy and integrity attacks against DP-3T; some of the attacks that we consider in this paper are indeed instantiations of the ones of Vaudenay, even though we give more emphasis to the possibility of exploiting such attacks for mass surveillance. The DP-3T team

reacted to Vaudenay's work by presenting a public response to his attacks [12] that does not object on their applicability, and mainly tries to convey the message that those attacks are hard to mount and inherent to any decentralized approach. Our results show that the there can be systems that are resilient to those attacks.

Pietrzak [18] proposed solutions and mitigations to replay and relay attacks against DP-3T. In such attacks, one can inject false notifications of at-risk contacts. Furthermore, Pietrzak identified the issue that users of DP-3T can easily provide digital evidence of contacts with infected users. Another way of injecting false notifications is via bribing of the infected users, such an attack was conjectured to be applicable to decentralized systems like DP-3T by Vaudenay [24]. Avitabile et al. demonstrated several concrete instantiations of such attack with respect to GAEN-based ACT systems [4]. Tang [21] observed that DP-3T may be subject to identification attacks and presented a comprehensive survey on proximity tracing systems. Canetti et al. provided general formalizations of ACT systems [8], including one in the Universal Composability framework. According to these formalizations, the DP-3T and GAEN systems lack basic privacy and integrity properties.

There are several works providing practical evidence of the above mentioned vulnerabilities. In [20], Seiskari showed a proof-of-concept implementation of Vaudenay's Paparazzi attack [23], demonstrating that, in DP-3T, locations visited by infected users during the two weeks prior to their diagnosis can be tracked by any third party (not only the government) that can install a large fleet of BLE-sniffing devices. Notice that such devices can be completely passive (i.e., they do not broadcast an identifier beacon) and therefore not easily detectable. Baumgärtner et al. [5] provided empirical evidence for two important risks of the Apple&Google design, namely, as in DP-3T, tracing of infected users and replay/relay attacks.

An alternative approach to the above decentralized systems consists of giving more power to a server. Examples of such more centralized ACT systems are TraceTogether[1], adopted in Singapore and ROBERT [16] adopted in France. In [14] the authors reviewed both centralized and decentralized ACT systems such as DP-3T, NTK, and ROBERT, analyzing the different adversarial models and the corresponding risks.

Finally, some works proposed alternative ACT systems trying to improve integrity. Pinkas and Ronen [19], building upon a design similar to DP-3T, proposed a system with an improved resilience to relay attacks and a better verification of risks. Buccafurri et al. proposed an alternative protocol that completely avoids the exchange of identifier over BLE and instead relies on GPS[7].

Public-key cryptography. Like Pronto-C2 some systems leveraged public-key cryptography to offer better privacy guarantees. One of such systems is WeTrace[2][9] where public keys are exchanged over the BLE channel. An infected user A uploads messages encrypted with the public keys related to close contacts she had. These encrypted messages are independent of the public key A was broadcasting while being in contact with the recipients of the messages. While this might be beneficial for A's privacy, it can make easier to mount attacks generating false notifications of at risk contacts. Indeed, a user B may be alerted consequently to an upload performed by an infected user C who did not come into contact at all with B, but just received B's key trough other means.

Inria published on Github the design of a system named DESIRE [15] that uses the Diffie-Hellman protocol for key exchange. After being tested positive, a user uploads data related to the encounters he had during the previous days. Such data is computed by hashing the shared key along with an information depending on which of the two users is creating the report. This guarantees that if two users A and B have been in close proximity, and both of them end up being positive to SARS-CoV-2, they will send two different values to the server, making it impossible for the server itself to infer that A and B have been co-located. Differently from Pronto-C2, the at-risk status for each user is computed on a central server.

II. From DP-3T to GAEN

BLE allows smartphones physically close to each other to exchange identifiers requiring a low battery consumption. Such communication mechanism avoids GPS technology and third-party devices like Wi-Fi routers or base stations of cellular networks. In a BLE-based privacy-preserving ACT system after a short period of time 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 both the announced and received pseudonyms. The central idea is that whenever a person is detected infected, smartphones that have been physically close to the smartphone of the infected person for a certain amount of time should show an alert to notify the potential at-risk contact. To realize this, the smartphone of the infected person should use the above database to 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, the above approach guarantees some degree of privacy through the unlinkability of the pseudonyms. Nevertheless, the risk that such systems can be abused to violate privacy remains a major concern and can affect their adoption.

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.

[2]We became aware of WeTrace only few months after first publishing our work on ePrint, informed by Adrienne Fichter.

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

One can trivially realize a decentralized BLE-based ACT system giving to the server the role of proxy that forwards to non-infected persons the pseudonyms of the infected individuals who decide to upload their pseudonyms[3] after being detected infected. Therefore, everyone, including the server, learns directly pseudonyms that have been used during the previous days by recently infected persons. On the other hand, the pseudonyms generated by smartphones belonging to non-infected persons are not uploaded to the server and remain visible only to whoever was physically close to those smartphones. Such decentralized systems seemingly have a potential to offer better privacy protection compared to known systems that use the centralized approach.

It is truly problematic to realize BLE-based privacy-preserving smartphone applications that can practically (in the sense of usability, battery consumption, and so on) work on (almost) all currently used BLE smartphones, unless some flexibility is allowed by Apple&Google through updates of iOS and Android. To this regard, Apple&Google have released updates of iOS and Android providing an application program interface for exposure notification (GAEN) [2]. The access to the API is restricted to specific smartphone apps authorized by Apple&Google (i.e., the apps selected by governments). Sadly, if one would like to implement a usable smartphone application (i.e., an app that runs in the background without battery drain on a very large percentage of the currently available smartphones) that rotates the BLE identifier beacon then it is hard to avoid the use of GAEN and therefore the app must use their approach for pseudonym generation and exposition, inheriting the related limitations/vulnerabilities.

This lack of flexibility generates some interesting consequences. First of all, the centralized approach is hard to implement since it relies on pseudonyms generated by the server and then advertised in the BLE identifier beacon by the smartphone. However, the generation of pseudonyms can only happen inside the smartphone when using GAEN. Such mismatch implies that the decision of Apple&Google makes harder to realize the centralized approach and any other different system (even an alternative decentralized one) that relies on generating BLE identifier beacons in a different way.

III. Privacy Issues in DP-3T Protocols

Starting with the attacks presented by Vaudenay [23], we discuss the privacy issues in DP-3T. In some of the attacks a government through its natural power controls (even partially) the server, the laboratories that detect infections and the national territory to violate privacy, possibly to collect more data, and use them in mass surveillance programs.

[3]The actual information uploaded is a seed that generates the pseudonyms.

A major privacy problem is that in DP-3T (and all analogue systems) one can be traced even when walking alone. Indeed, a passive antenna can detect a pseudonym without transmitting anything (we name this *silent tracing*), 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 has been using. Such antennas can be installed nearby any place where the citizen can be identified, and this allows to connect pseudonyms to identities. Concretely, the key weakness of DP-3T is that asking smartphone applications to hand over the used keys/pseudonyms to the server is like asking infected citizens to kneel down in front of the Big Brother.

IV. A Decentralized Alternative: Pronto-C2

As main contribution of this work, we present Pronto-C2, a new decentralized privacy-preserving ACT system based on BLE. Pronto-C2 can be implemented through government servers but also can be fully decentralized using blockchain technology. Full decentralization can play an important role in ACT systems since many citizens may prefer to use their smartphones only when systems are transparent and resilient to integrity attacks, in addition to being privacy preserving.

Our decentralized solution relies on a paradigm shift compared to the approach of DP-3T. 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.

Diffie and Hellman proposed a key exchange protocol (i.e., the DH protocol) where two parties can establish a secret key K by just sending one message each on a public channel. 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. 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 use K in order to call 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 identifier 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 ideally it should be implemented through a decentralized blockchain so that we can decentralize the server, making the entire process transparent and reliable.

When implementing the bulletin board with a blockchain, the verification of the authorization must be performed by a smart contact and thus the check should be accomplished uniquely with public information. For this reason, we suggest the use of digital signatures. To make the upload of K unlinkable with the real identity of the infected person, we suggest the use of blind signatures. The basic idea is that

laboratories receive from the government some unpredictable activation codes that are then one by one given to infected persons. Then, an infected person connects to a service in order to exchange the authorization code with some blind signatures that will be useful to then upload on the bulletin board data associated to calls. In case of use of a blockchain to implement the bulletin board, this exchange of an authorization code with a blind signature is performed off-chain since the server will use a signature secret key, and thus it cannot be directly implemented by a smart contract.

Notice that the approach of Pronto-C2 is completely different from the one of DP-3T. Indeed, while in DP-3T the pseudonyms of the infected person are broadcast to everyone, 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 similar to a phone call where the infected party sends to the answering party the following message:[4] "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 download the recently uploaded keys to search for K (data don't need to be stored, the search can happen while downloading data). Notice that there is a different key K to check for every BLE identifier beacon received in the last two weeks that has not been already discovered. This step should be preferably performed while the phone is connected to the charger and to a Wi-Fi network. Moreover, for those cases where the daily amount of data to download is excessive, one could specify target states/regions in the country, so that only a restricted amount of information needs to be managed. In this case, a call would also specify a corresponding state/region. In addition to K, the infected person can also upload some auxiliary information (e.g., about BLE signal) to improve risk scoring or to share data with epidemiologists.

We remark that avoiding that two smartphones with pseudonyms A and B upload the same K (this would leak some –most likely irrelevant – 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 to use 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 identifier beacon. One might think to resolve the issue of the small space in a BLE identifier beacon by just resorting to very short (and therefore in our view too risky in case of mass surveillance attacks) keys or by splitting the information into multiple identifier beacons that rotate quickly. We instead propose a different approach that allows to use many bits for the group element while still remaining with one identifier beacon only. We decouple the group element from the pseudonym precisely like in operating systems a large amount of data is represented by a pointer. Recall that a value announced in a BLE identifier 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 identifier 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. Notice that this generation of group elements is done only once in a while, and therefore can typically be performed when the smartphone is on charge and is connected to a Wi-Fi network.

Silent tracing. Pronto-C2 is clearly secure with respect to silent tracing. In fact, it is based on virtual anonymous calls originated from a recently detected infected person, and addressed to whoever has been in close proximity with her. 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. In fact, there will be no key K that can be found in the list of virtual anonymous calls.

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 such attack is easier to detect compared to silent tracing. Indeed, the smartphone application could easily inform the owner at any time on the number of BLE identifier beacons that are currently received. Therefore, there is more room for citizens to realize the existence of malicious devices and ask police to destroy them and to identify the criminals that were trying to abuse the ACT system. Any government that would like to save its reputation convincing citizens to still use the smartphone application should take severe actions against such criminals. Obviously, if there is no prompt reaction of the government then citizens will feel that some attempts of mass surveillance are in progress and will simply switch off the smartphone application.

Pronto-C2 is secure against shameless tracing. Indeed, even though with shameless tracing the attacker will receive calls from an infected user, the calls are not linkable and the infected citizen remains hidden among all other infected citizens.

Side-channel attacks. As in all ACT systems, users could be de-anonymized 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 artificial delays and uploads of bogus data with the only purpose to confuse and make harder to achieve any profiling attempt. Furthermore, we suggest also a simple solution to mitigate such linkability problems which consists of allowing each user to select her own favorite mixer. Such mixer can be selected among several options that can belong to heterogeneous entities (e.g., political parties, large organizations defending civil rights). By doing so, users could pick their favorite options to protect their IP addresses when uploading their pseudonyms and their anonymous calls

[4] "Pronto" stays for "Hello" and C2 pronounced in English stands for "it is you" in Neapolitan language, as in the title of a song by Nino D'Angelo.

to the bulletin board. Moreover, the user can send batches of pseudonyms and calls since they will be mixed by the mixer that will also apply some artificial delays and dummy traffic, thus guaranteeing some level of unlinkability. We give a more detailed description of this idea in Section V. We stress that ACT systems currently deployed are at least in part affected by such issues and somehow ignore them. Still, we prefer to discuss possible workarounds, even though they obviously introduce extra overhead. DoS attacks can be mitigated with standard approaches (e.g., CAPTCHAs, proofs of work, anonymous tokens).

Remark on the "Paparazzi" attack of [23] and the DP-3T answer [12]. Vaudenay in [23] showed a privacy attack to DP-3T proposing an antenna that can be used to eavesdrop the identifier beacons sent by smartphones. The DP-3T team answered to [23] in [12] claiming that "This is a known attack vector inherent to all contact tracing systems, whether centralized or decentralized (SRE, Inherent Risk 1)". Our ACT system Pronto-C2 contradicts this claim. It might be that the DP-3T team was implicitly referring only to systems that follow their decentralized approach. This would imply that Apple&Google provides through GAEN a system that is inherently affected by non-inherent attacks to privacy.

V. Performance of Pronto-C2

In this section we describe how Pronto-C2 could be used in concrete scenarios and analyze its performance. We remark that one should not think that an ACT system must perform well in all possible scenarios in order to be considered. The sole fact that in some countries the performance of an ACT system is good enough makes the system of practical relevance and worthy to study. Pronto-C2 could be a viable solution in some countries, but it might not be efficient enough in others.

Pronto-C2 involves the following actors: the user U, who runs the smartphone application; the server Server, that manages the bulletin board; the medical laboratory HA; the authentication service AuthService.

There is a risk that U is subject to linkability/deanonymization attacks due to timings and IP addresses of the TLS connections with Server when uploading or downloading data. Such attacks also affect DP-3T, and in general are applicable to any system if no specific countermeasure is used. One might consider onion routing and mix networks to protect U against such attacks, but the impact on performance remains unclear. In order to give a fair description of a practical realization of Pronto-C2, we do not ignore this issue and we therefore include here a mitigation based on mixers. We will consider a setting where U can freely select a mixer MixServer that she trusts, and mixers do not need to be approved by the government, they can be spontaneously run by anyone.

Server owns a pair of private and public keys $(\mathsf{sk}_{\mathsf{Server}}, \mathsf{pk}_{\mathsf{Server}})$ of a public key encryption scheme (e.g., ElGamal instantiated on the elliptic curve used for the key exchange), the public key of Server is made publicly available at set-up time. Every time U has to send data to Server, U will actually encrypt the data with $\mathsf{pk}_{\mathsf{Server}}$ and send the resulting ciphertexts to MixServer. A mixer waits for enough data to be collected, and then performs a mixing and sends them to Server. In addition, MixServer can also download all data from Server so that U can use MixServer also to retrieve anonymous calls and ephemeral keys.

There can be several heterogeneous mixers available, provided by large institutions like no-profit organizations, political parties, national/state/local governments, as well as several smaller mixers that can serve a district, a school, a group of friends/relatives. U will obviously choose the one that he trusts more in performing properly the service with a sufficiently large amount of collected data and without abusing it. It remains possible for a user to ignore this suggestion and to use some proper delays and then sending the encrypted data directly to Server through onion routing and/or relying on the partial hiding provided by mobile operators and public Wi-Fi networks (somehow they can also be seen as light forms of mixers). We will continue our discussion considering the case of a citizen using a mixer that she trusts.

Server works as a bulletin board, so all data ever received by Server are made publicly available after being decrypted. HA is the laboratory that perform the SARS-CoV-2 tests. If a user U gets tested positive, HA hands U an authorization code Code. AuthService is the service in charge of authorizing users to upload anonymous calls to the bulletin board. It can also be useful to issue credentials to upload ephemeral keys to the bulletin board, in order to mitigate DoS attacks.

Each user U executes the following operations preferably when the smartphone is connected to a battery charger, and preferably having Internet access via Wi-Fi[5]:

- U generates a set of 96 ephemeral and secret keys to be used for the next day (there is a rotation every 15 minutes).
- U connects to AuthService to prove to be a legitimate user of the system needing to announce new pseudonyms. With this connection, U obtains 96 blind signatures of the generated ephemeral keys. This step is needed to avoid DoS attacks on the bulletin board.
- U uploads the ephemeral keys to the bulletin board. U encrypts the 96 ephemeral keys, along with the related unblinded signatures, and sends the resulting ciphertexts to MixServer that, after having collected a sufficiently large amount of data, mixes and sends them to Server who will eventually decrypt and publish them. U will obtain the addresses of his ephemeral keys by querying Server with the first l bits (the value of l will be discussed later) of each key. Server will return, for each query, all the ephemeral keys that match these bits, along with the addresses of such keys. U will store the addresses of his ephemeral keys and will broadcast them during the following day. By doing so, U is able to efficiently retrieve his addresses while hiding the link between them to Server since each query corresponds to a fairly large

[5]We implicitly assume that all TCP/IP connections use TLS.

set of ephemeral keys. To add more noise, dummy queries may also be performed.

- U downloads from Server (this step could also be performed through MixServer that can have a local copy of data available on Server) the ephemeral keys $\mathsf{Eph}_{\mathsf{U}_i}$ for all addr_i collected during the day by querying Server (or MixServer) with the first l bits of each address collected during the day. For each query, U will receive a set of ephemeral keys and U can select the needed key.
- U downloads from Server (as above this step could also be performed through MixServer) all the calls that have been added to the bulletin board after U performed the last download of the calls (e.g., the previous day). Consequently, U evaluates his contagion risk.
- If U is positive to SARS-CoV-2, U receives Code from HA who delivered him the diagnosis.
- The infected user U computes the anonymous call for each ephemeral key received, optionally excluding the ones encountered at a specific time or date, and connects to AuthService to obtain a blind signature for each call. This is done by asking AuthService to sign one call at a time. Notice that the government knows the identities of infected citizens, therefore there is no need of special protection in this step.
- The infected user U encrypts the calls along with the unblinded signatures, with $\mathsf{pk}_{\mathsf{Server}}$. Then, U uploads these data via MixServer, as done for the upload of the ephemeral keys. If U feels uncomfortable in giving evidence of being infected to MixServer, U can send dummy encrypted calls on a daily basis (i.e., whenever U uploads new ephemeral keys).

Note that Pronto-C2 will behave exactly as in DP-3T regarding data sent during the day over the BLE channel (e.g., it will send and receive the same amount of data without performing additional computations).

To give an idea of the overall performance, we report an example of a concrete execution in a typical scenario. To analyze the performance, we take into account the memory usage of the smartphone application, the amount of uploaded and downloaded data, and the number of exponentiations the smartphone has to execute. We assume that: each user U has 100 contacts per day on average; there are on average 5000 new infected individuals per day within a single country; U uses a new pseudonym every 15 minutes; the contagion time window is 10 days long; the dummy calls produced by the user are 100 per day on average; there are 5 million users that upload their ephemeral keys each day; we set $l = 17$. By doing so the resulting set of ephemeral keys would be of about 3663 elements on average. [6]. In the same scenario, if l is equal to 10, the number of ephemeral keys that is downloaded by each user to compute the anonymous calls is 468750 on average, while, increasing l to 25 the number of ephemeral keys downloaded is 15. In general, the number of downloaded keys is the total number of ephemeral keys published divided by 2 to the number of bits fixed in the prefix. Compared to l equale to 15, in case l is equal to 10 the requests of the user are hidden in a larger number of calls, while in case l is equal to 25 the number of ephemeral keys downloaded by U is smaller and it is easier for an adversary to guess which ephemeral key was searched by U.

Every smartphone has to maintain on the local memory less than 150KB. We did not count the space required by blind signatures since they are sent to MixServer and erased as soon as they are received from AuthService.

By using blind signatures based on RSA [6] with a length of 2048 bits for the modulus, the size of downloaded data is about 177MB[7]. The vast majority of daily downloaded data comes from the anonymous calls needed to evaluate the infection risk and their size is about 153MB, that can be processed as soon as received and deleted immediately after that, without flooding the memory of the smartphone. Downloading 177MB could be expensive in terms of rate plan, therefore we recommend to perform this operation when the smartphone is connected to a Wi-Fi connection. Furthermore, governments, cooperating with mobile operators, could reduce or eliminate such costs. In this way, users who do not have access to a Wi-Fi connection would be able to benefit from the service anyway, obviously limiting the use of MixServer to the steps that are more critical for privacy protection.

The amount of daily uploaded data is less than 350KB if the user is not infected. An infected user uploads also about 250KB daily.

The smartphone computes one exponentiation for each blind signature. In order to get an ElGamal encryption the smartphone computes 2 exponentiations for each 32 bytes of plain-text data.

On daily basis there are about 2000 exponentiations for dummy calls and the total number of exponentiations needed to store all the ephemeral keys is around 1100.

Also there are daily 100 exponentiations to risk of contagion. Therefore, the total number of exponentiations computed on average per day by a smartphone of a non-infected user is below 3500. If the user is tested positive to SARS-CoV-2, the number of additional exponentiations he performs is 10000.

VI. Attacks, Analysis and Comparison

We consider the following attacks.

Paparazzi attack: Adv controls a fleet of passive BLE devices (i.e., they receive messages but do not transmit) placed in locations that are known to him. Adv traces the movements of the infected users during the contagion time window.
This attack is not applicable to Pronto-C2 since different ephemeral keys (i.e., pseudonyms) generated by the same user U are not linkable. Adv could try to track a target infected user U exploiting the calls available on the bulletin board. Since the

[6] l can be chosen dynamically considering the number of ephemeral keys that have been added that day and ensuring that the size of the set of keys allows for an efficient search while providing sufficient privacy.

[7] Even if the amount of downloaded data can seem considerable, it is worth noting that when using a video streaming application (e.g., YouTube), smartphones typically download very large amounts of data.

devices used by Adv are passive, no calls of U will ever be directed to Adv. The only way for Adv to track U is to extract the ephemeral keys used to generate the calls and associate them to a single user. Since the calls are anonymously sent to Server, Adv fails in linking together the calls of U.

Orwell attack: the attack is analogous to Paparazzi but Adv now also receives all data that is in possession of the Server. This attack is not applicable to Pronto-C2. Assuming that the upload of the calls on the bulletin board is performed through an anonymous channel, in order to link these calls, Adv needs to discover which call was blinded by U before obtaining the corresponding signature from AuthService. This would require Adv to break the blindness property of the blind signature. Even if Adv breaks the blindness of the signature scheme, the additional information received is the set of calls sent to Server by U, but none of them is a call that Adv understands. Indeed, none of the passive devices controlled by Adv is the recipient of a call. Adv would need to take all the pairs of ephemeral keys recorded by each passive device D_i, and try to compute a call between the two users that owns these two ephemeral keys. If the computed call is equal to a call published by U, then Adv knows that U was located in proximity to D_i. However, if Adv is able to successfully compute a call starting from two ephemeral identifiers, it is easy to show that Adv can be used to define an adversary breaking the DH assumption.

Matrix attack: Adv is as in the Orwell attack in terms of the information he has access to. On the other hand, Adv's devices can actively send messages over the BLE channel. Adv combines data in his possession with the ability to actively send messages of the contact tracing protocol over the BLE channel in order to trace infected citizens over the contagion time window. Since all the uploaded calls are made unlinkable to each other, thanks to the blindness of the signature scheme and the use of anonymous channels, even if Adv is the recipient of certain calls, Adv is in general unable to link all the calls generated by the same user U. Obviously, there are some inherent leaks in extreme situations like when there is only one new infected person in the area where pseudonyms have been collected by the adversary. Such leaks are seemingly inherent and Pronto-C2 provides the strong resilience against shameless tracing.

Bombolo attack: Adv consists of Server and HA colluding together. When users are tested positive, they upload data to the system. Adv uses such data to compute additional information about infected users beyond the data they reported, such as the number of their contacts and co-location information among other infected users.
Pronto-C2 is vulnerable to this attack. Indeed, the number of calls to AuthService sent by U is clearly exposed. We note that this leak of information can be mitigated through dummy calls. It is important to note that co-location information among infected users is not leaked since two infected users who met each other will upload uncorrelated data.

Brutus attack: this attack allows Adv to link the pseudonyms of a user with his real identity. Adv is colluding with the server and the health authorities and tries to discover the real identity of the users uploading data to the server.
This attack fails with Pronto-C2. The data uploaded by a user U in Pronto-C2 cannot be linked to the real identity of U. Data are uploaded in the following steps:

1) when the infected user interacts with AuthService in order to obtain the blind signatures of the calls, and
2) when the infected user uploads the calls to Server along with the unblinded signatures.

The first step involves uploading the authorization code Code to AuthService in order to obtain the blind signature of the calls. Since HA knows the real identity of each infected user, it is possible for Adv to link the blind signature requests with an infected person. However, since the upload of the calls is performed through an anonymous channel, Adv cannot link the calls with the signature requests thanks to the blindness property of the signature scheme. This of course hides the identity behind the uploaded data only inside the set of the infected users, which is known to Adv.

Gossip attack: Adv has the same capability as a regular user of the system. The attack is successful if Adv can produce plausible digital evidence of an encounter with an infected user. The lack of resilience to the Gossip attack can be also seen as a potential feature since overwhelmed laboratories could prioritize requests for tests of citizens who can present a reasonable proof of contact with an infected user. At first sight, one could think that a proof of contact with an infected user U can be given by user providing a proof about the calls on the bulletin board. For instance, let A be a user holding a secret key sk_A corresponding to Eph_A and let Eph_U be the ephemeral key of a user U. If A finds a call $K = H(\mathsf{Eph}_\mathsf{U}^{\mathsf{sk}_\mathsf{A}}||\mathsf{Eph}_\mathsf{U}||\mathsf{Eph}_\mathsf{A})$ on the bulletin board, A could prove that he knows the secret key sk_A corresponding to Eph_A and that K is computed as before, thus proving that U made a call to A. However, in Pronto-C2, all the pseudonyms used by the users are made public on the bulletin board. So, even if A was never in contact with U, A could use Eph_U, that is public on the bulletin board, to compute a call $K = H(\mathsf{Eph}_\mathsf{U}^{\mathsf{sk}_\mathsf{A}}||\mathsf{Eph}_\mathsf{U}||\mathsf{Eph}_\mathsf{A})$ (a call from U to A) and show sk_A as proof of the fact that such call has been done. Generally, any proof of the fact that U made a call to A is *not* evidence of the fact that U met A since such proof could have been computed by A even if U was never in contact with A. Notice, however, that in the case A is not infected, what we have just described is instead plausible evidence of the fact that U met A: indeed, only infected users can write calls to the bulletin board and U is honest (if U is dishonest, the pair U and A can be seen as a single adversary). For this reason, we say that the attack affects Pronto-C2 minimally in the sense that an attacker A can provide a proof (that, as shown before, is the secret key sk_A) of the contact between U and A that convinces a third party B who believes that A is not infected.

Matteotti attack: Adv attempts to produce false alerts by causing non-at-risk users to get notified of a risk. The adversary can collude with the server and the health authorities and place passive BLE devices at locations of his choice.
This attack is not applicable to Pronto-C2. Every call K stored

Attacks	Low-cost DP-3T	Unlinkable DP-3T	Pronto-C2
Paparazzi	✗	✓	✓
Orwell	✗	✗	✓
Matrix	✗	✗	✓
Bombolo	✓	✓	✢
Brutus	✗	✗	✓
Gossip	✗	✗	✢
Matteotti	✓	✗	✓
Replay	✗	✓	✓

Figure 1. Identified attacks. We show which system is susceptible to which attack. ✗ denotes that the system is vulnerable to the attack, ✓ safety against the attack and ✢ minimal impact from the attack.

on the bulletin board has the form $K = H(K'||\mathsf{Eph_C}||\mathsf{Eph_B})$. A user B who at some time t broadcasts $\mathsf{Eph_B}$ will be notified a risk only if B received at time t an ephemeral key $\mathsf{Eph_C}$ and $K' = \mathsf{Eph_C^{sk_B}}$. Since it is hard for Adv to compute K' without knowing $\mathsf{sk_B}$ or $\mathsf{sk_C}$, we conclude that B is alerted only when B actually met C and C put an alert for B. However, in such case the alert corresponds to an actual risk for B and does not represent a successful attack.

Replay attack: Adv collects pseudonyms at a location X where the probability to meet an infected person is high and broadcasts such pseudonyms to users at a different location Y. The listened pseudonyms are broadcast at a later time slot. This attack is not applicable to Pronto-C2. An adversary Adv who broadcasts, at location X, the pseudonym of a user U_1 collected during a prior time slot in a different location Y, would fail in the attempt of causing false at-risk notifications. Indeed, to be notified, a user U_2 needs to find, on the bulletin board, a call which is directed to himself and is generated by the infected user U_1. Since the generation of such call requires the secret key of U_1 related to the time slot when the alleged meeting took place, it would be computationally infeasible for Adv to trigger a fake at-risk notification for U_2. In Figure 1 we compare Pronto-C2 with DP-3T in relation to the above attacks. The second column of the table refers to a protocol proposed by the DP-3T team in which the server computes a Cuckoo filter containing the pseudonyms of the infected users.

Acknowledgments

The full version of this paper can be found on ePrint [3].

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), by region Campania (POR) and by national funds (PON).

References

[1] "TraceTogether - behind the scenes look at its development process," https://www.tech.gov.sg/media/technews/tracetogether-behind-the-scenes-look-at-its-development-process.

[2] Apple and Google, "Privacy-Preserving Contact Tracing," *https://www.apple.com/covid19/contacttracing*.

[3] G. Avitabile, V. Botta, V. Iovino, and I. Visconti, "Towards defeating mass surveillance and SARS-CoV-2: The Pronto-C2 fully decentralized automatic contact tracing system," IACR ePrint Arch., Report 2020/493, 2020, https://eprint.iacr.org/2020/493.

[4] G. Avitabile, D. Friolo, and I. Visconti, "TEnK-U: Terrorist attacks for fake exposure notifications in contact tracing systems," IACR ePrint Arch., Report 2020/1150, 2020, https://eprint.iacr.org/2020/1150.

[5] L. Baumgärtner, A. Dmitrienko, B. Freisleben, J. Höchst, M. Mezini, M. Miettinen, T. D. Nguyen, A. Penning, F. Roos, A.-R. Sadeghi, M. Schwarz, and C. Uhl, "Mind the GAP: Security & privacy risks of contact tracing apps," in *TrustCom 2020, Security Track*, 2020, pp. 458–467.

[6] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, "The one-more-rsa-inversion problems and the security of chaum's blind signature scheme," *J. Cryptol.*, vol. 16, no. 3, pp. 185–215, 2003.

[7] F. Buccafurri, V. D. Angelis, and C. Labrini, "A privacy-preserving solution for proximity tracing avoiding identifier exchanging," in *Proc. of International Conference on Cyberworlds*. IEEE, 2020, pp. 235–242.

[8] R. Canetti, Y. T. Kalai, A. Lysyanskaya, R. L. Rivest, A. Shamir, E. Shen, A. Trachtenberg, M. Varia, and D. J. Weitzner, "Privacy-preserving automated exposure notification," IACR ePrint Arch., Report 2020/863, 2020, https://eprint.iacr.org/2020/863.

[9] A. D. Carli, M. F. Franco, A. Gassmann, C. Killer, B. Rodrigues, E. J. Scheid, D. Schoenbaechler, and B. Stiller, "Wetrace - A privacy-preserving mobile COVID-19 tracing approach and application," *CoRR*, vol. abs/2004.08812, 2020.

[10] J. Chan, L. P. Cox, D. P. Foster, S. Gollakota, E. Horvitz, J. Jaeger, S. M. Kakade, T. Kohno, J. Langford, J. Larson, P. Sharma, S. Singanamalla, J. E. Sunshine, and S. Tessaro, "PACT: privacy-sensitive protocols and mechanisms for mobile contact tracing," *IEEE Data Eng. Bull.*, vol. 43, no. 2, pp. 15–35, 2020.

[11] DP-3T's Team, "Decentralized Privacy-Preserving Proximity Tracing," *https://github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf*.

[12] DP-3T's Team, "Response to Analysis of DP3T: Between Scylla and Charybdis," https://github.com/DP-3T/documents/blob/master/Security%20analysis/Response%20to%20'Analysis%20of%20DP3T'.pdf.

[13] L. Ferretti, C. Wymant, M. Kendall, L. Zhao, A. Nurtay, L. Abeler-Dörner, M. Parker, D. Bonsall, and C. Fraser, "Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing," *Science*, vol. 368, no. 6491, 2020.

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

[15] Inria PRIVATICS Team, "DESIRE: A Third Way for a European Exposure Notification System," https://github.com/3rd-ways-for-EU-exposure-notification/project-DESIRE/blob/master/DESIRE-specification-EN-v1_0.pdf, 2020.

[16] Inria PRIVATICS Team, "ROBERT: ROBust and privacy-presERving proximity Tracing," https://github.com/ROBERT-proximity-tracing/documents/blob/master/ROBERT-specification-EN-v1_0.pdf, 2020.

[17] 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.

[18] K. Pietrzak, "Delayed authentication: Preventing replay and relay attacks in private contact tracing," in *Proc. of Progress in Cryptology - INDOCRYPT 2020 - Lecture Notes in Computer Science*, vol. 12578. Springer, 2020, pp. 3–15.

[19] B. Pinkas and E. Ronen, "Hashomer - a proposal for a privacy-preserving bluetooth based contact tracing scheme for Hamagen," https://github.com/eyalr0/HashomerCryptoRef/blob/master/documents/hashomer.pdf, 2020.

[20] O. Seiskari, "Contact Tracing BLE sniffer PoC," https://github.com/oseiskar/corona-sniffer, 2020.

[21] Q. Tang, "Privacy-preserving contact tracing: current solutions and open questions," *CoRR*, vol. abs/2004.06818, 2020.

[22] TCNCoalition, "TCN Protocol," https://github.com/TCNCoalition/TCN#the-tcn-protocol, 2020.

[23] S. Vaudenay, "Analysis of DP3T," IACR ePrint Arch., Report 2020/399, 2020, https://eprint.iacr.org/2020/399.

[24] S. Vaudenay, "Centralized or decentralized? The contact tracing dilemma," IACR ePrint Arch., Report 2020/531, 2020, https://eprint.iacr.org/2020/531.