Digital Contact Tracing Solutions: Promises, Pitfalls and Challenges

Thien Duc Nguyen[1], Markus Miettinen[1], Alexandra Dmitrienko[2], Ahmad-Reza Sadeghi[1], and Ivan Visconti[3]

[1]*Technical University of Darmstadt, Germany - {ducthien.nguyen, markus.miettinen, ahmad.sadeghi}@trust.tu-darmstadt.de*

[2]*JMU Würzburg, Germany - alexandra.dmitrienko@uni-wuerzburg.de*

[3]*University of Salerno, Italy - visconti@unisa.it*

***Abstract*—The COVID-19 pandemic has caused many countries to deploy novel digital contact tracing (DCT) systems to boost the efficiency of manual tracing of infection chains. In this paper, we systematically analyze DCT solutions and categorize them based on their design approaches and architectures. We analyze them with regard to effectiveness, security, privacy and ethical aspects and compare prominent solutions based on these requirements. In particular, we discuss shortcomings of the Google and Apple Exposure Notification API (GAEN) that is currently widely adopted all over the world. We find that the security and privacy of GAEN has considerable deficiencies as it can be compromised by severe large-scale attacks.**

We also discuss other proposed approaches for contact tracing, including our proposal TRACECORONA, that are based on Diffie-Hellman (DH) key exchange and aim at tackling shortcomings of existing solutions. Our extensive analysis shows that TRACECORONA fulfills the above security requirements better than deployed state-of-the-art approaches. We have implemented TRACECORONA and its beta test version has been used by more than 2000 users without any major functional problems[1], demonstrating that there are no technical reasons requiring to make compromises with regard to the requirements of DCT approaches.

***Index Terms*—digital contact tracing, privacy, security**

I. INTRODUCTION

The pandemic caused by the SARS-CoV-2 corona virus has still the world in its grip since it was officially announced by the World Health Organization (WTO) on March 11, 2020. At the time of writing, we have been witnessing the surge of several infection waves all around the world. Reliable and efficient contact tracing for containing the spread of infections has therefore become more important than ever. In many countries, digital contact tracing apps on smartphones have already been rolled out to support manual contact tracing with the hope of significantly improving its effectiveness in breaking infection chains and preventing the virus from spreading further. In this paper, we focus on analyzing how theoretical results of epidemiologists (e.g., [1]) are taken into account in current proposals for identifying at-risk contacts in the presence of technological errors, data pollution attacks and privacy and ethics regulations. Initially we analyze deployed solutions, as many countries are currently actively employing them and millions of users are affected by such systems. Regardless of the potential usefulness of digital contact tracing or a lack thereof, contact tracing apps have become a reality in many countries. At the time of writing, 49 countries around the world (including, e.g., most European countries, Australia, China, Singapore) and 27 states in the USA have deployed contact tracing apps[2]. Many of these systems in use today were designed, implemented and rolled out in great haste with the goal of containing the spread of the pandemic as quickly as possible. It is therefore ever more important to take a step back and try to obtain a critical view of the benefits and disadvantages of individual approaches.

In this context, *effectiveness*, *security*, *privacy* and *ethics* are key aspects that need to be considered thoroughly: (i) the system should be *effective*, i.e., able to provide acceptable detection accuracy (high true positive and low false positive rate), (ii) it should be *secure* so that malicious adversaries cannot manipulate the system to trigger false alarms, (iii) it should protect *privacy* to increase users' trust in the DCT system, and (iv) it should consider ethical aspects as it should be transparent and based on voluntary use. Ensuring all above properties is necessary to achieve high adoption rates to then significantly contain the spread of the virus. Otherwise, users will not be willing to use contact tracing apps, negatively impacting their adoption rate that would be crucial for their effectiveness in practice (ideally higher than 60%) [2].

While the first countries (predominantly in Asia) that deployed tracing apps adopted centralized approaches, and extensively collected sensitive user information (e.g., names, addresses, mobile phone numbers, location), a widespread and heated debate on user privacy broke out in Europe and the USA[3]. In this turmoil of evolving contact tracing approaches, Google and Apple established an unprecedented collaboration and provided their special application programming interface for decentralized contact tracing called *Exposure Notification*

[1]https://tracecorona.net/download-tracecorona/

[2]MIT Covid Tracing Tracker, https://tinyurl.com/3ey44r5c

[3]In the course of this debate about 300 security and privacy researchers from 26 countries signed an open letter criticizing the specific privacy risks of some centralized contact tracing approaches, advocating privacy-preserving solutions whenever better privacy can be obtained without penalizing effectiveness (https://drive.google.com/file/d/1OQg2dxPu-x-RZzETlpV3lFa259Nrpk1J/view). This signed letter has been often abused claiming that centralized systems are bad and decentralized systems do what is needed to detect at-risk contacts, and moreover they do it protecting privacy.

API (GAEN) [3] which they rapidly integrated into their mobile operating systems. Google and Apple give in each country access to this interface only to one organization that is authorized by the local government. GAEN runs an almost complete contact tracing solution as a part of the underlying mobile operating systems, so that the role of national organizations is reduced to developing a user interface to GAEN through a smartphone app and providing the backend server infrastructure required for acquiring and distributing information about at-risk contacts. Further, although Apple and Google initially promised not to get directly involved in contact tracing by developing their own backend server and app, later they did so by providing the GAEN Express solution that is used in several US states, e.g., Maryland and Utah[4]. Unfortunately, it is known that existing rolled out Digital Contact Tracing (DCT) systems exhibit a number of important security and privacy risks [4], [5], [6], [7].

In order to tackle the shortcomings of existing approaches, we introduce a novel user-controlled privacy-preserving contact tracing system called TRACECORONA. It leverages a robust privacy architecture based on Diffie-Hellman key exchange to provide a level of security and anonymity unparalleled by any of the other systems proposed so far. It also improves the effectiveness and accuracy of the overall system and its resilience to misuse through the ability to *verify* all critical encounters.

In particular, we provide following contributions:

- We introduce a categorization of the requirements on DCT systems in four dimensions, namely: effectiveness, privacy, security and ethical considerations (Sect. III).
- We propose a novel distributed contact tracing system based on Diffie-Hellman (DH) key exchange, TRACECORONA, providing strong security and privacy guarantees (cf. Sect. IV). In contrast to almost all existing approaches that are based on exchanging pseudonymous proximity identifiers, our approach leverages advanced cryptographic algorithms to establish and verify encounter tokens that are unique to each encounter between two users. Futher, we propose various use cases and deployments of TRACECORONA including a hybrid approach (cf. Sect. IV-D). We implemented, deployed, and published TRACECORONA for beta user test (cf. Sect. IV-E).
- We analyze TRACECORONA in comparison to prominent schemes w.r.t these aforementioned requirements (cf. Sect. V). Our analysis shows that DH-based systems provide better security and privacy guarantees than GAEN while maintaining comparable effectiveness.

In summary, we provide a comprehensive set of requirements to evaluate DCT systems. We show that current approaches do not fulfill such requirements at large, e.g., have number of security, privacy and effectiveness issues. Hence, we propose TRACECORONA, a novel approach that address the deficiencies of existing DCT systems. In the following, we will present those requirements of DCT systems as well as TRACECORONA in details. Further, we have published a full version of this paper as a technical report that includes a systematization and extensive analysis of existing DCT schemes as well as the extended application scenarios of TRACECORONA [8].

[4]MD COVID Alert, https://tinyurl.com/yeymtrm2

TABLE I: Notations.

User (U)	A Person that uses a DCT App
User App (App)	A DCT app installed on users' devices
Tracing Service Provider (SP)	Providing a system (e.g., servers and apps) for identifying at-risk contacts
Health Authority (HA)	Authenticating the user infection status
Infected user	A user that has tested positive for COVID-19
Affected user	A user that has encountered an infected user
Indirect contacts	A user that has encountered an affected user

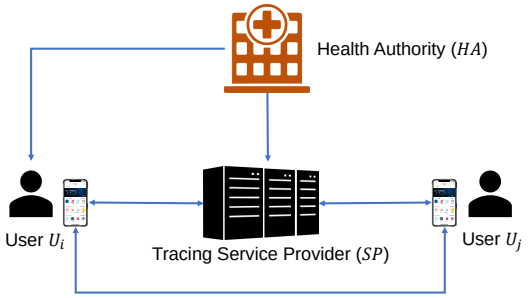


Fig. 1: System model of Digital Contract Tracing (DCT).

II. DIGITAL CONTACT TRACING

In this section, we present the system model, architectures and technologies of DCT systems.

A. System Model

Figure 1 shows the typical system model of contact tracing schemes. There are three types of entities: Users U (e.g., U_i and U_j) of the tracing system (app), a contact tracing Service Provider (SP), as well as a health authority (HA). In the following, we discuss these roles in more detail.

1) Users: A user U_i uses a dedicated *contact tracing app* installed on its device (typically a smartphone) to collect information required to determine contacts with other users of the system. Different technologies can be used for this purpose, e.g., directly through exchange of specific information over a proximity communication protocol like Bluetooth LE, or, indirectly with the help of a trace of location information obtained from a positioning system like GPS, by determining simultaneous co-presence of the users at the same location at the same time. We will discuss various technologies in Sect. II-C. Users' contact tracing apps collect and store this information about contacts of users locally on users' mobile devices. In case a user U_i is tested positive with a disease (like COVID-19), the user is expected to use the contact tracing app to warn other users of the system by uploading the collected information about his/her contacts to the contact tracing service provider SP.

2) Tracing Service Provider: The Tracing Service Provider SP is responsible for collecting and distributing information necessary for identifying contacts with infected users and/or notifying other users of such contacts. In *centralized* systems,

the *SP* determines contacts between infected users and other users and issues notifications to them, whereas in *decentralized* systems, the determination of possible contacts is performed by the users' contact tracing apps.

3) Health Authority: The Health Authority *HA* is responsible for identifying infected users (e.g., through administered medical tests) and authenticating their infection status towards *SP*. This is necessary to prevent malicious users $\mathcal{A}^u$ from pretending to be infected and thereby triggering false alarms with users they have had contacts with. To do this, *HA* will issue a user-specific unique authenticator, e.g., a transaction authentication number (TAN) (a form of single use one-time password (OTP)) to an infected user U_i, who can subsequently present this authenticator when uploading their information to *SP*. By verifying the authenticator with *HA*, the *SP* can verify the infection status of the user U_i.

B. Centralized vs. Decentralized Architectures

In general, contact tracing approaches can be divided into two main design architectures, *centralized* and *decentralized*, based on whether the identification of encounters between users is performed by the tracing service provider *SP* or by the tracing apps of users *U*. Both approaches are based on individual users' tracing apps recording temporary identifiers (*TempID*s) of other devices they encounter. In the case a user U_i is infected, he uses his tracing app to upload identifiers to *SP*. In centralized systems, the recorded identifiers of *other* apps will be uploaded, whereas in decentralized systems, the *TempID*s used by the tracing app *itself* in the recent past will be uploaded. The main difference between these schemes is the fact that in the centralized system the service provider *SP* generates all *TempID*s centrally and is therefore able to link the infected user with the (pseudonymous) identities of other users, whereas in the decentralized approach, the *TempID*s are generated individually by each tracing app. The determination of contacts can therefore only be performed by the actual tracing apps involved in an encounter. The tracing app conducts this by downloading the *TempID*s of infected users, e.g., U_i from *SP* and comparing these to the *TempID*s the tracing app has encountered in the past. This approach therefore effectively limits the exposure of sensitive information about encounters to *SP*.

In contrast to common belief, however, this difference does not directly guarantee "privacy by design" for decentralized systems and susceptibility to "mass surveillance" in centralized systems. The actual evaluation of these models highly depends on the underlying architectural decisions and on the various threat models considered.

Due to space constraints, we refer the reader to Sect. IV of our technical report [8] for a systematization and discussion of state-of-the-art contact tracing schemes.

C. Technologies to Determine Encounters

In general, there are two types of technologies to determine encounters: (1) location-based technologies such as GPS and QR-codes used for venue check-ins and (2) peer-to-peer proximity detection-based technologies like Bluetooth, Ultra-wideband (UWB), and ultrasound. Currently, Bluetooth is the most dominant technology deployed in contact tracing. Therefore, in the following, we focus on Bluetooth technology and refer the reader to Sect. II.B of [8] for the detailed discussion of other technologies.

Bluetooth Low Energy (BLE). BLE can be used for sensing the proximity between individual users' devices, e.g., [3], [9]. Indeed, many recent approaches for contact tracing on smartphones use Bluetooth proximity detection. The participating smartphones beacon out information like temporary identifiers (*TempID*s) that can be sensed by other devices. In addition, also related metadata like the signal strength of the beacon may be recorded. Using the signal strength information, some approaches seek to provide estimates about the distance of the encounter. However, it has been shown that signal strength can provide only a very rough estimate about the actual distance of devices, as it is influenced by other factors like device orientation and surrounding structures [10]. Nevertheless, since BLE is widely available on most recent smartphone versions, it seems the most viable alternative for implementing proximity detection on smartphones that are widely used by the population in many countries.

Compared to GPS and QR-code based approaches, BLE would seem to reveal the least amount of information about the users because *HA* and *SP* do not collect physical locations as well as actual encounter times. Thus, only anonymized random strings are shared among the apps using BLE. However, BLE-based approaches still have several security, privacy, effectiveness, and ethical problems. For example, they are susceptible to fake exposure injection attacks, e.g., relay attacks, or user profiling, e.g., movement tracking and user identification. We will elaborate all of these problems in detail in Sect. V.

III. Requirements for DCT Systems

As mentioned above, digital contact tracing (DCT) schemes need to collect information about infected individuals. Although many countries have deployed contact tracing apps, the effectiveness of DCT is so far still unclear. Moreover, DCT poses a number of privacy and security challenges on the underlying scheme design, since it collects and processes sensitive information which is related to users' health and users' contacts to some extent. In this section, we systematically consider the requirements for DCT based on four pillars: effectiveness, privacy, security, and ethical aspects. These requirements are broken down and listed in Tab. II. Next, we will discuss each of them in detail.

A. Effectiveness

In the following, we discuss three sub-requirements for the effectiveness of a DCT system, namely, *Accuracy*, *Superspreader*, and *Accountability*.

1) Accuracy (R-Ef1): For accurately estimating the risk of contagion it is necessary to estimate the duration of each contact (in minutes) along with a good estimate of the distance between the users involved in the encounter. The duration of contacts ideally could be detected by continuously scanning

for the presence of BLE devices in proximity to verify the continued presence of other devices. This aggressive approach will, however, lead to significant energy consumption draining the smartphone battery quickly. In practice, one needs therefore to pause the scanning for several seconds before the next scan to preserve energy. Computing a good estimate of the distance between devices is even more challenging since there are multiple factors (e.g., positioning of the antenna in the smartphone, obstacles in between smartphones, and their orientations) that introduce significant errors to distance estimates. Indeed, experiments performed by Leith and Farrel [10] showed that GAEN is quite imprecise in estimating the distance of devices of potential at-risk exposures.

2) Superspreaders (R-Ef2): The mere capability of detecting at-risk exposures was initially considered sufficient by many endorsers of decentralized systems like, e.g., the team around the influential DP-3T [11] contact tracing approach, which also had a considerable influence on the GAEN design adopted by Google and Apple. However, along the way, more epidemiological insights about the behavior of SARS-CoV-2 have been discovered. Among them is the fact that a very relevant aspect for understanding the spread of the virus is the important role of so-called *superspreaders*. Indeed, Reichert et al. [12] showed that while there is a large percentage of infected individuals that do not transmit the virus at all, there is a small fraction of infected individuals that instead are very contagious and cause numerous further infections. A DCT system aiming at effectively defeating SARS-CoV-2 should therefore also take into account the importance of superspreaders and provide mechanisms allowing to detect them and their potential contacts.

Contagious asymptomatic infected individuals (CAIIs). Particularly problematic are so-called asymptomatic infected individuals, i.e., persons that are infected and contagious, but asymptomatic and thus may unwillingly spread the disease. Such individuals have a very low chance of being tested positive since they do not show any symptoms of being sick and therefore will not likely seek to be tested. Even if they want to be tested, in many countries, they will not be prioritized in testing. Hence, they can have an active role in spreading the virus. However, as such individuals are unlikely to be tested and receive a positive diagnosis from *HA* (which is a prerequisite for uploading information about contacts to the service provider *SP*), it is unlikely that such persons will ever be able to use the DCT system to warn other users about possible at-risk contacts with them.

3) Accountability (R-Ef3): Implementing, deploying, and operating a DCT system can be very costly and requires a majority of the population to participate in its operation. Therefore, the system should provide adequate and valid information about its effectiveness in a privacy-preserving way. For example, the system should be able to provide basic statistics about the number of active users, infected users, users notified about potential at-risk exposures, as well as false positive rates, etc. At a minimum, the system should be able to demonstrate clear benefits in comparison to a purely random selection of users to be quarantined in specific at-risk groups (e.g., where the infection rate is higher) [10]. Although some GAEN-based apps do provide reports on some measures related to the system's effectiveness, such measures can be biased, unreliable or misleading [13], [14] as we will discuss in Sect. V.

B. Privacy

The main privacy concerns relate to the abuse of a DCT in order to *identify* users, *track* users, or *extract the social graph* of users. Information that is emitted to the user's surroundings by contact tracing apps and shared with other involved parties should not introduce such privacy risks as elaborated next.

1) Identifying users (R-P1): DCT systems aim at identifying encounters, not users. Therefore, the systems should not leak any information that can be used to establish the true identity of any individual user.

2) Tracking users (R-P2): DCT apps work by continuously beaconing pseudonymous identifiers into their surroundings. These identifiers should not be linkable, i.e., it should not be possible to trace the movements of any user over time, as this may potentially enable to deduce facts about the user's behaviour and lead to an identification of the user.

3) Extracting the social graph (R-P3): In general, contacts (especially long encounters), are often related to social relationships (i.e., users that decide to be close to each other). When handling contact information, a DCT system should make sure that one cannot abuse information collected by it to generate a relevant part of the social graph of any user, since this may enable to draw conclusions about social relationships between users and thus potentially identify them.

Note: Obviously, there exists in some cases inherent information leakage due to specific circumstances, e.g., in situations in which the adversary is in the proximity only to one specific person. If the adversary later receives an at-risk notification, it will be trivial for the adversary to conclude that this one person is indeed the infected person. Therefore, when considering the above three privacy requirements, we will always focus on *large-scale attacks* and will in particular focus on identifying attacks affecting potentially many users.

C. Security

The effectiveness of a DCT system is severely impacted if a system is not resilient to large-scale data pollution attacks. Such attacks can generate, for instance, false at-risk notifications (false positives) therefore jeopardizing the correctness of the contact tracing system. Indeed, massive false at-risk notifications could result in spreading panic among the general population. Moreover, this could also cause unnecessary strain on the health system through unnecessary testing and negative impact on the society due to unnecessary self-quarantining.

1) Fake exposure claims (R-S1): The system should prevent a malicious or dishonest user $\mathcal{A}^u$ that aims to circumvent the DCT system to claim that he or she has encountered an infected user. There can be different motivations for this attack: (i) $\mathcal{A}^u$ aims to harm the reliability of the system by manipulating encounter checking results, (ii) $\mathcal{A}^u$ uses the fake exposure status as an excuse to stay at home instead of going to work or participating in an event, and (iii) $\mathcal{A}^u$ intentionally

shares wrong encounter information to epidemiologists, thus sabotaging their analysis of the epidemiological situation.

2) Fake exposure injection - Relay/replay attacks (R-S2): This attack aims to inject fake contacts on a large scale resulting in many false exposure notifications. Here, a fake contact indicates the state that the DCT system incorrectly determines that two users were in "close contact" at a specific time although they were not. It affects the main goal of DCT system as to identify contacts that potentially cause high exposure risks. Relay attacks are a typical example of fake exposure injection attacks. In a relay attack, the adversary captures the temporary IDs of a user U_i and broadcasts them in other locations (e.g., other cities). As a result, the system incorrectly identifies the users in the other locations who captured those temporary IDs to have encountered U_i.

D. Ethics

1) Transparency and voluntary participation (R-Et1): The whole process (design, development, deployment, and operation) of a contact tracing system must be transparent to users and the systems must be removed immediately when the pandemic is over to avoid misuse. Further, users should be free to decide whether they want to participate in the system or not, and be free to withdraw their participation anytime they wish. Otherwise, users will not trust, and thus will not be willing to use DCT apps. This will affect the crucial need of a high adoption rate of DCT.

2) Independence (R-Et2): The contact tracing process (design, development, deployment and operation) in a particular region should be independent of any parties with potential vested interests. Procedural controls of the contact tracing system should underlie a transparent public scrutiny and be solely under the control of democratically-elected governments. In particular, giant technology corporations (e.g., Mobile OS vendors) should not be allowed to use their technological or market dominance to control or drive DCT systems since they might be biased in it for the sake of their own subjective benefits, e.g., using DCT data for business purposes could undermine the de-facto ability of legitimate governments to oversee the use of data collected for contact tracing purposes.

TABLE II: List of requirements for digital contact tracing.

	Requirement	Description
	Effectiveness	
R-Ef1	Accuracy	Specifying distance and duration of encounters
R-Ef2	Superspreader	Identifying superspreaders and their contacts
R-Ef3	Accountability	Providing statistics to evaluate the actual effectiveness
	Privacy	
R-P1	Identifying users	Users should always remain anonymous
R-P2	Tracing users	Users should not be tracked
R-P3	Extracting social graph	Making sure that no social graph can be extracted
	Security	
R-S1	Fake exposure claim	Preventing malicious users to lie about their exposure status
R-S2	Fake exposure injection	Preventing relay/replay attacks
	Ethics	
R-Et1	Transparency and voluntary use	The system must be transparent and based on voluntary use
R-Et2	Independence	Ones should not be allowed to use their technological or market dominance to control DCT systems in their favour

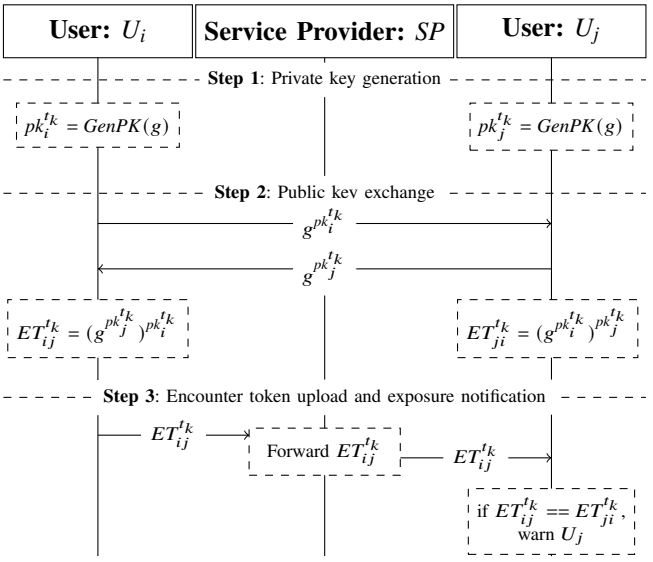


Fig. 2: Generic framework of DH-based Approaches.

IV. Proposed Approach - TraceCORONA

In this section, we first provide a generic framework for Diffie-Hellman (DH)-based schemes. We then present our novel scheme, TraceCORONA, a fully fledged example of a DH-based approach and highlight its benefits compared to the prominent approaches analyzed in Sect. V.

A. Generic framework of DH-based approaches.

The core idea of decentralized approaches based on asymmetric key cryptography like Diffie-Hellman is that two users establish a *unique and secret* Encounter Token (ET) using a key exchange protocol when they are in proximity by exchanging short-lived random public keys via BLE. In this paper, we use Diffie-Hellman as a key exchange protocol. Figure 2 shows an overview of the use of DH-based encounter tokens in a contact tracing scheme. In **Step 1**, users U_i and U_j generate their own private keys $pk_i^{t_k}$ and $pk_j^{t_k}$ respectively for each time interval t_k that is changing every T (e.g., 15) minutes. These private keys are used to derive corresponding public keys $pubk_i^{t_k} = g^{pk_i^{t_k}}$ and $pubk_j^{t_k} = g^{pk_j^{t_k}}$. In **Step 2**, the public keys are exchanged via BLE when two devices are in vicinity. For encounters surpassing a specified minimal duration, e.g., 5 minutes, an ET will be calculated, e.g., U_i calculates $ET_{ij}^{t_k}$ from U_i's private key $pk_i^{t_k}$ and U_j's public key $pubk_j^{t_k}$ as follows: $ET_{ij}^{t_k} = (g^{pk_j^{t_k}})^{pk_i^{t_k}}$. Since U_i and U_j never share their private keys, only they can know their secret encounter token $ET_{ij}^{t_k}$. It is worth noting that the DH key generation and encounter token calculation processes do not

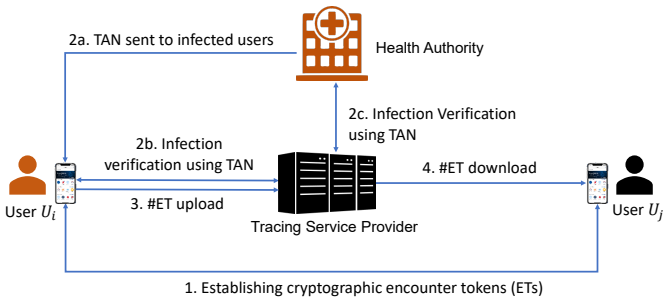


Fig. 3: TRACECORONA system overview.

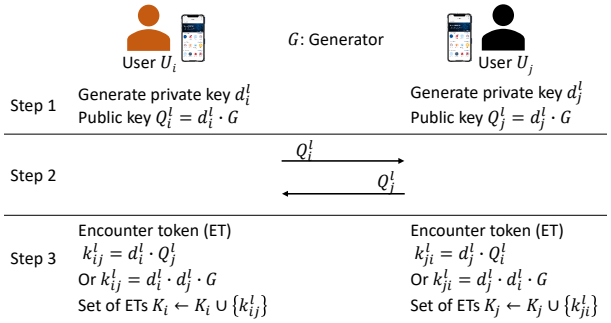


Fig. 4: Elliptic-curve Diffie–Hellman (ECDH)-based encounter token establishment.

need to happen on-line. For saving battery, it can be deferred to the next time when the smartphone is being charged. In **Step 3**, when a user (e.g., U_i) is tested positive for COVID-19, U_i sends its encounter token $ET_{ij}^{t_k}$ to the *SP* which will forward $ET_{ij}^{t_k}$ to other users. Once U_j receives $ET_{ij}^{t_k}$, it will compare $ET_{ij}^{t_k}$ to the *ET*s it has calculated. If $ET_{ij}^{t_k}$ is equal to $ET_{ji}^{t_k}$, U_j is notified that it has encountered an infected user.

Although we use the well-known DH-based approach for illustrative purposes, any other two-party key-exchange protocols where parties send only one short message to each other are applicable. Thus, existing proposals like CleverParrot [15], PRONTO-C2 [16], and Epione [17] use Elliptic-curve DH (ECDH). Further, these approaches provide several modifications and optimizations to improve the effectiveness, security and privacy of the system (cf. Sect. VI-A).

B. Limitations of DH-based approaches

Our proposed approach TRACECORONA seeks to address three technical limitations of DH-based approaches as follows:

- **Size restriction of BLE beacon message.** Since public keys are in general too big for BLE beacon messages, existing solutions apply workarounds, e.g., PRONTO-C2 needs to handle a bulletin board, or CleverParrot has to reduce the key size and requires operating systems to enable special BLE advertising messages.
- **Sharing encounter tokens *ET*s.** Uploading *ET*s directly may raise privacy risk. Hence, we aim to keep *ET*s always secret.
- **No time window restriction.** Existing approaches do not limit limit time window that would open opportunity for two-way relay attacks.

In the following, we will present TRACECORONA and discuss how we address those limitations in detail.

C. TRACECORONA Design

1) System Overview: Our design follows the system model (cf. Fig. 1) and the generic framework for DH-based schemes shown in Fig. 2. An overview of the basic usage scenario of TRACECORONA is shown in Fig. 3. For a discussion on complementary application scenarios like wearable devices and private contact tracing please refer to Appendix C of [8].

The functionality of TRACECORONA can be divided into four phases: (1) Encounter token establishment, (2) infection verification, (3) token information upload, and (4) token information download and contact verification. Next, we will describe each of these phases in detail.

2) Encounter Token Establishment: TRACECORONA App uses BLE as a proximity communication protocol to advertise a random ephemeral identifier to other devices in the environment and to scan for the identifiers of other apps. Once an ephemeral identifier of another app has been observed for a minimum duration (e.g., 5 minutes), a connection over BLE to the other app is opened and an Encounter Token (*ET*) is established using the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol. Figure 4 shows the token establishment protocol in detail for two users U_i and U_j. Following typical ECDH notation, let Q denote the public key, d the private key and G the generator. Let T denote the period of a rolling key time frame and l be the index of the time frame $f^l = [l*T, (l+1)*T]$. Let K_i and K_j be the sets of ETs of users U_i and U_j, respectively. Let k_{ij}^l be an ET established between two user Apps U_i and U_j at time point t_{ij}^l, i.e., a timestamp falling in time frame f^l. The process of establishing an ET is then as follows:

1) **Step 1:** For every time frame f^l, users U_i and U_j generate a ECDH keypair including private keys d_i^l and d_j^l, and public keys $Q_i^l = d_i^l * G$ and $Q_j^l = d_j^l * G$, respectively, where G is the generator defining the used cyclic subgroup of the elliptic curve.
2) **Step 2:** U_i and U_j exchange their public keys Q_i^l and Q_j^l via Bluetooth LE.
3) **Step 3:** Each user calculates the encounter token based on its private key and the received public key. In particular, U_i calculates $k_{ij}^l = d_i^l * Q_j^l$ while U_j calculates $k_{ji}^l = d_j^l * Q_i^l$. Obviously, $k_{ij}^l = k_{ij}^l = d_i^l * d_j^l * G$. Each user then adds the encounter token into its encounter token set: $K_i \leftarrow K_i \cup \{k_{ij}^l\}$ for U_i and $K_j \leftarrow K_j \cup \{k_{ji}^l\}$ for U_j. After k_{ij}^l is established, U_i and U_j continue exchanging their ephemeral identifiers periodically to monitor the duration D_{ij}^l of the encounter and the strength of the Bluetooth signals S_{ij}^l (which roughly correlate with how far or near two users are from each other). In summary, the data recording the start of the encounter t_{ij}^l, the duration of the encounter $D_{k_{ij}^l}$ and the strength of the Bluetooth signal S_{ij}^l, are stored as metadata associated with token k_{ij}^l.

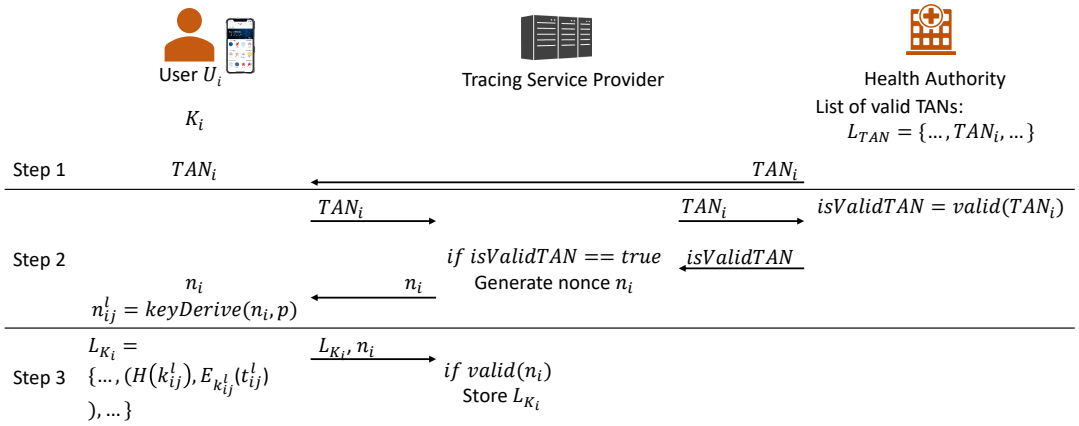
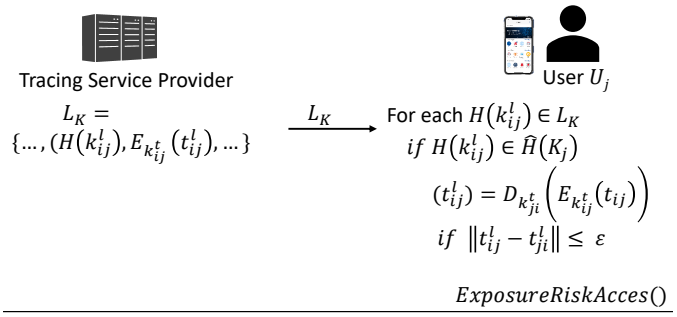


Fig. 5: Infection verification and encounter token upload.



L_K : The set of shuffled hashes and metadata of all encounter tokens of the infected users from the last update.

Fig. 6: Encounter Token Download and Exposure notification.

It is worth noting that in order to preserve battery lifetime, **Step 1** and **Step 3** can be done offline, i.e., when the smartphones are being charged (e.g., during the night).

3) Infection Verification and Encounter Token Upload: Since the main goal of the system is to notify users who have encountered infected users (tested positive for COVID-19), the system needs to make sure that only infected users can use the system to release their encounter tokens K. In our system, the Health Authority *HA* issues for each infected user a unique authentication code, a so-called Transaction Authentication Number (*TAN*). If an infected user wants to share their encounter tokens, it can use this *TAN* to prove its infection status by uploading the *TAN* along with their encounter token information.

Fig. 5 illustrates the infection verification and encounter token uploading phases. **In Step 1** and **Step 2** *HA* sends a TAN_i to infected user U_i. This can be done by using any appropriate out-of-band channel: in person, via SMS, via regular mail or via e-mail. TAN_i can also be sent along with the test results. The infected user can input their *TAN* directly by typing the number in or use their smartphone's camera to scan a QR code containing the *TAN*. **Step 3** shows how U_i can upload its encounter token information. Timestamp t_{ij}^l is encrypted using AES encryption using the encounter token k_{ij}^l as the key (or a key derivation function can be used to derive a key from k_{ij}^l). Let $m_{ij}^l = E_{k_{ij}^l}(t_{ij}^l)$ denote the encryption of t_{ij}^l. U_i sends TAN_i and a list L_{K_i} consisting of the m_{ij}^l along with corresponding hashes $h_{ij}^l = H(k_{ij}^l)$ of the encounter tokens k_{ij}^l to server *SP*. We have thus $L_{K_i} = \{\ldots, (m_{ij}^l, h_{ij}^l), \ldots\}$. *SP* forwards TAN_i to *HA* to verify whether TAN_i is valid or not. If TAN_i is valid, it will extract and store each element (m_{ij}^l, h_{ij}^l) of L_{K_i} separately.

It is worth noting that TRACECORONA provides both usability and privacy benefits by enabling infected users to remove specific unnecessary or sensitive encounter tokens that, e.g., (1) had only a short duration, thus being not essential for contracting the disease, or, (2) happened at a time or place that users do not want to disclose even anonymously, e.g., at a sensitive event or meeting.

4) Encounter Token Download: All TRACECORONA Apps download regularly, e.g., every night, encounter token information from server *SP* to identify potential exposure risks. Figure 6 shows the encounter download protocol. Let $L_k = \{\ldots, (H(k_{ij}^l), E_{k_{ij}^l}(t_{ij}^l)), \ldots\}$ be the list of the hashes and metadata of all encounter tokens of all infected users since the last update. To avoid linking entries related to a particular infected user together based on their position in the list, all entries in L_k are shuffled before sending them to users. Once a user U_j receives L_k, it compares the received token hashes to its own token hashes to discover matching encounters. If a matching encounter hash, e.g., $H(k_{ij}^l)$ is identified, U_j decrypts the matching encounter token metadata using the associated encounter token k_{ij}^l as the key: $t_{ij}^l = D_{k_{ji}^l}(E_{k_{ij}^l}(n_i||t_{ij}^l))$. U_j then checks the validity of encounter token w.r.t. to encounter time t_{ij}^l to make sure that k_{ij}^l and k_{ji}^l were established during the same time frame. This will limit the time-window available for a relay attack as we will discuss in Sect. V-C. Assuming that the clocks of the two devices are deviating by at most ϵ seconds, if $|t_{ij}^l - t_{ji}^l| \leq \epsilon$, k_{ij}^l and k_{ji}^l are considered to have been derived at the same time, i.e., the matching of k_{ij}^l and k_{ji}^l is valid. The system then uses metadata information, e.g., the time of the encounter t_{ij}^l, the duration of the encounter $D_{k_{ij}^l}$ and the signal strength S_{ij}^l to assess the risk of this exposure.

TABLE III: Useful information for epidemiological analysis and evaluation and optimization of a DCT system.

Number of active users
Number of infected users
Number of encounters of infected users
Number of affected users
Number of encounters of affected users
True positive rate
Importance of notification [5]
Distribution of risk score
The correlation between risk score and true positive rate

D. Hybrid Approach

In the following, we will present a hybrid approach that provides a trade-off between the effectiveness and the privacy requirements of centralized and decentralized architectures, i.e., maximizes effectiveness of the app while preserving privacy of the users. As discussed in Sect. III-A3, the accountability requirement (R-Ef3) refers to the possibility to evaluate the effectiveness of a DCT scheme. Therefore, we focus on this requirement by specifying what kind of data are needed to satisfy it and how they can be submitted to the health authority *HA* and the tracing service provider *SP*.

1) Useful data: To fulfill the requirement R-Ef3 (Accountability), the App needs to send authentic, but anonymized data in a privacy-preserving way to *SP*. Table III shows potentially useful types of data that can help to evaluate and optimize the DCT system. Such types of data can also be helpful to epidemiologists and decision makers to understand the virus spreading patterns and, e.g, deploy effective policies to limit the pandemic.

2) Sharing Epidemiological Information with Health Authorities: As discussed in the previous section, a direct contact U_j can prove it exposure status with an infected user (U_i) based on the possession of the secret value of the encounter token ET_{ij}. TRACECORONA utilizes this to authenticate the correctness of exposure information that users may voluntarily want to share with health care research institutions, thereby preventing malicious users from corrupting the data by providing faked exposure information to the researchers. This helps in improving the accuracy and correctness of the epidemiological modelling used as as basis for political decision making in the crisis situation.

3) Sharing Epidemiological Information via Healthcare Professionals: Since healthcare professionals like doctors collect information about their patients that come for a COVID-19 test or for consultation for their symptoms, doctors can act as a source of reliable information for epidemiological analysis in a properly anonymized form. For example, the healthcare professional could provide for each patient following anonymous information to help in assessing the epidemiological situation as well as the effectiveness of the contact tracing system: whether the user was notified by the contact tracing app and what the possible risk score was, whether the user knew about a potential exposure status even before being notified by the app, possible symptoms, and the test result. These kind of data provided to the epidemiological analysis do as such not reveal any information about the true identity of individual patients, but they do provide crucial information necessary to evaluate the effectiveness of the contact tracing app.

[5]Exposure notification from a DCT is less important if users already knew their exposure status before being notified by a DCT app, e.g., the affected users who live in the same household to an infected users are expected to be informed immediately when the test result is available.

E. Implementation and Beta Test

We prototyped TRACECORONA for the Android smartphone platform and tested it in a public beta test. We have not implemented TRACECORONA on iOS because it does not allow apps to use Bluetooth communication in the background [18]. We use the native Android BLE APIs to implement the Encounter Token Establishment protocol. Further, our cryptographic functions, e.g., ECDH are based on the Bouncy Castle library. For the server acting as *SP*, the code is written in Java and run on Ubuntu Server operating system. In principle, our app can run on any Android smartphone that supports Bluetooth LE, i.e., Android 5.0 and later.

Alpha testing. We internally tested the app with 25 devices covering various models and manufacturers. The results show that our app works without any problems and consumes 5 to 8% battery for a whole day (24 hours) of contact tracing without further optimizations.

Beta test. We published the TRACECORONA app on our website and interestingly the app has drawn a lot of attention[6]. Indeed, more than 2000 users have downloaded and tested the app. We have received many positive feedbacks on the app features and performance, except received criticism that the app does not work on very old devices that do not support Bluetooth LE. However, this is a technical limitation that is out of our control.

Implementation on wearable devices. To demonstrate the possibility of deploying TRACECORONA even on on wearable devices like wristbands a MCU developer board that costs about US $20 (For a full description please refer to Appendix C of our full technical report [8]), we have implemented our design on Adafruit HUZZAH32 (ESP32).

V. SECURITY AND PRIVACY ANALYSIS OF TRACECORONA

In this section, we will analyze DH-based approaches in general and TRACECORONA in particular in comparison to GAEN and BlueTrace with regard to requirements laid out in Sect. III. Due to space constraints, we refer the reader to Sect. V of our full technical report [8] for detailed discussion on the shortcomings of state-of-the-art contact tracing schemes including BlueTrace [9] and GAEN [3].

A. Effectiveness

Accuracy. As discussed in Sect. III-A1, measuring the distance between smartphones using BLE is not very reliable due to its the inherent technical limitations. Hence, we note that all approaches based on BLE-proximity sensing share the same challenge of not being able to reliably estimate the distance between devices involved in a contact. Therefore, none of

[6]https://tracecorona.net/download-tracecorona/

BLE-based approaches can entirely fulfill the Accuracy requirement R-Ef1. One potential solution to increase distance measuring accuracy could be using BLE in combination with other sensors like ultra-wideband (UWB) (cf. Sect. II-C of [8] for the details).

Superspreader. Although the Tracing Service Provider *SP* only receives anonymous encounter tokens that are not sufficient to detect Superspreaders and CAII users, the contact tracing App itself can be used to warn its user in case the App identifies a large number of contacts with other infected users, since this can be an indication that actually the user itself is a Superspreader or CAII who has been the source of contagion for those infections. As a result, the user could seek immediate testing, but also immediately upload their encounter tokens to warn others. Further, the App can prove the user's status as a suspected Superspreader or CAII to *SP* by uploading the secret encounter tokens it has in common with infected users. By verifying these against the published hashes of encounter tokens of infected users the *SP* can verify that the user is indeed a person with many contacts with infected people and therefore a possible Superspreader. The *SP* can then tag the encounter tokens of the user accordingly, so that exposure notifications related these tokens can additionally be marked as being related to a 'possible superspreader' contact. Hence, requirement R-Ef2 related to the ability to identify Superspreaders can be successfully addressed.

B. Privacy

In DH-based systems, the public keys change every 15 minutes. This means that an eavesdropper adversary $\mathcal{A}^e$ cannot link public keys of a user, i.e., $\mathcal{A}^e$ can only track the movement of a user for less than 15 minutes, which is not enough to build informative movement profiles of the user.

Surveillance. Like other decentralized BLE systems, this attack fails against DH-based systems since the matching of contacts is done exclusively by the Apps. A malicious service provider $\mathcal{A}^s$ does not benefit from learning the ETs of infected users since the uploaded encounter tokens do not reveal any information about the counterparts of those encounters.

Mass Surveillance. In TRACECORONA, even if a malicious service provider $\mathcal{A}^s$ colludes with an eavesdropper $\mathcal{A}^e$, the adversaries only get to know the hashes of encounter tokens of infected users and possible locations where $\mathcal{A}^e$ has collected them. However, as discussed in Sect. IV-B, since $\mathcal{A}^e$ can obtain *ET*s only through direct interaction with the monitored users and *ET*s are created only if encounters last for a specific time (e.g., 5 minutes), $\mathcal{A}^e$ is much more limited in its ability to obtain *ET*s associated with other users. In particular, $\mathcal{A}^e$ will be unable to establish *any ET*s with users that are just shortly passing by an eavesdropping station, so that the adversary's ability to track the movements of infected users is very limited. It is to be noted that this is a significant difference existing approaches (cf. Sect V of [8]), since in these approaches the ability of the eavesdropping adversary $\mathcal{A}^e$ is in this sense unlimited and it can effectively sense the presence *all* users passing by its eavesdropping stations, even based on *one single observation* of the user.

In the case of malicious service provider $\mathcal{A}^s$ (i.e., the service provider *SP* is dishonest), $\mathcal{A}^s$ could link encounter tokens ETs of a specific infected user since the tokens would be submitted in one transaction when they are uploaded to the service provider *SP*. One solution to prevent this threat is to apply appropriate anonymization (privacy) techniques, e.g., blind signatures with an anonymous postbox service [19] or private set intersection [17] to the upload process of encounter tokens. We discuss such advanced privacy techniques in details in Appendix B of [8]. In particular, these techniques minimize the risks that neither malicious service provider $\mathcal{A}^s$, health authority *HA* nor any party can link individual encounter tokens of infected users, thereby limiting the trackability of individual users to relatively short time frames of, e.g., 15 minutes. Therefore, by applying such techniques, TRACECORONA can effectively address the requirements regarding providing protections against identifying (R-P1) and tracking (R-P2) users and extracting their social graphs (R-P3).

C. Security

Next, we will explain how DH-based systems can mitigate current attacks, hence, fulfill the security requirements.

Fake exposure claim. DH-based systems can mitigate fake exposure claims (requirement R-S1). As mentioned in Sect. IV, infected users only share the hashes of encounter tokens meaning that the values of the encounter tokens themselves are always kept secret, so that only users actually participating in the encounter obtain the corresponding encounter token. Therefore, by proving possession of the (secret) encounter token, a user can prove that a contact with the counterpart has in fact taken place. The only way a dishonest user $\mathcal{A}^u$ can make fake exposure claims is to obtain access to the phones of users having matching encounter tokens and extracting them. However, this attack requires compromising individual devices one-by-one and hence cannot be easily scaled.

Relay/Replay Attacks. These attacks aim to inject false exposure notifications *on a large scale*. Unfortunately, widely adopted approaches like BlueTrace and GAEN are vulnerable to various relay attacks [5], [13], [7], [20], [21]. For example, Baumgärtner et al. [5] have demonstrated a real-world relay attack on GAEN in two cities (Frankfurt and Marbug) in Germany. They show that the adversary can capture and relay $tempID$s among those cities. They estimate that the attack can inject about 76 $tempID$s from infected users to a mobile device within 15 minutes. Principally, all proximity-based approaches are vulnerable to such attacks. However, DH-based systems provide two effective mitigation techniques that reduce the window of opportunity for attackers: (i) *two-way communication* is required for establishing contact tokens, prohibiting massive abuse by just copying and broadcasting beacon information, and (ii) using *limited time windows* for validating the timestamp of an encounter.

Two-way communication. In contrast to existing approaches [3], [11], [22], [23], [9] that are vulnerable to *one-way* relay attacks (cf. Sect. V [8]), DH-based schemes utilize a *handshake* protocol requiring *two-way* communication to establish an encounter token. This means $\mathcal{A}^w$ cannot simply capture

the beaconed information in one place and broadcast it in many other places like it would be possible in other schemes. $\mathcal{A}^w$ has to capture and relay messages at both places at the same time. This not only limits the time window of the attack but also imposes a restriction on the scale of the attack since a mobile device cannot communicate with too many other devices at the same time due to the limited number of channels and bandwidth that Bluetooth LE provides. Based on our estimation, an average smartphone can only handle 8 Bluetooth LE connections simultaneously in a reliable manner. Therefore, in theory $\mathcal{A}^w$ can relay the handshake of one device to at most 8 other remote devices, while this number is not limited in other approaches.

Limited time window. In DH-based schemes, two users U_i and U_j in proximity of each other establish a unique secret encounter token ET_{ij}. An infected user U_i can use ET_{ij} to encrypt any meta-data that only U_j can decrypt. Leveraging this property, in a DH-based scheme, e.g., TRACECORONA, the exact timestamp of an encounter can be encrypted and added to encounter token metadata so that user Apps checking encounter tokens can also check the exact encounter time. Therefore, only matching encounters that took place within a time window of at most ϵ seconds are considered as valid encounters, thereby limiting the window of opportunity for relay attacks. Other decentralized schemes like [3], [11], [23] cannot impose such limitations on the timestamps of ephemeral IDs, because the involved tracing apps can not mutually verify the actual time point of when contacts take place due to the fact that only one-way communication is used. Due to this, the GAEN API [24] allows a two-hour time window for synchronizing *RPI*, i.e., $\mathcal{A}^w$ can have up to two hours to conduct relay attacks. In DH-based schemes, this ϵ could be limited to seconds when assuming that smartphones used for contact tracing apps can sync their clocks via an Internet connection or during the exchange of the public keys. Note that all contact tracing apps need a frequent Internet connection for uploading and downloading encounter information.

Therefore, the combination of these two advantages, requirement of two-way communication and small time window help DH-based schemes such as TRACECORONA to significantly reduce the impact of relay attacks on the system.

D. Ethics

Like BlueTrace, DH-based systems like TRACECORONA can be implemented with complete access to the source code, guaranteeing transparency. It is a standalone app that does not depend on any built-in contact tracing APIs running deep inside the mobile operating systems such as Android or iOS, thus satisfying requirements with regard to transparency and (R-Et1) and independence (R-Et2). This is in stark contrast to proprietary and closed GAEN systems strictly enforced by Google and Apple. Especially in Apple's iOS systems independent contact tracing applications that continuously need to use BLE in the background are blocked by the operating system so that effective BLE sensing as required by contact tracing apps is in practice not possible. Instead, Apple forces all contact tracing approaches to rely on their closed and proprietary GAEN API whose functionality can not be independently examined nor verified. It is therefore highly debatable, whether this approach is ethical, as Apple in fact forces users into using their GAEN solution, having to involuntarily accept all possible related deficiencies, or, refrain from using contact tracing solutions at all. One solution to make DCT systems independent from mobile OS vendors w.r.t BLE and GAEN APIs is to use third-party wearable devices as discussed in detail in Appendix C of [8].

E. Summary of Benefits of DH-based Approaches and Comparison to Other Approaches

We summarize key differences and security and privacy advantages of DH-based systems in comparison to existing approaches in Tab. IV. As can be seen in the table, GAEN does not fulfill the requirements. The DH-based systems provide better security and privacy protection than all other discussed solutions. For example, DH-based approaches are resilient to fake exposure claim attacks and wormhole adversary (i.e., narrowing the attack window time and requiring more communication effort as the adversary would have to operate real-time two-way communication relays). Moreover, comparing to the most widely spread contact tracing framework by Apple and Google, which is vulnerable to profiling attacks as the adversary can track the movements of infected users, DH-based systems guarantee a better protection. Interesting but not surprisingly, BlueTrace is the best w.r.t to fulfilling effectiveness requirements since it can potentially detect Superspreader and CAII and provide useful data to epidemiologists while this could be challenging to other approaches. In terms of ethics, GAEN again is on the lower end because it received many criticisms due to their coercion and the lack of transparency. More importantly, our hybrid approach inherits the advantages of DH-based approaches in terms of security and ethical aspects, while being on par with centralized approaches with regard to effectiveness.

VI. RELATED WORK

A. DH-based approaches

PRONTO-C2 [16]. The main problem of DH-based approach is that the size of the public key might exceed the space limit of BLE advertising messages. The minimum requirement for a standardized ECDH key is 256 bits (or 384 bits to provide security against a powerful adversary) while in a typical BLE advertising message there is space for 128 bits only. PRONTO-C2 stores the public keys on a bulletin board that can be maintained by the *SP* or can be decentralized, and implemented with a blockchain. Hence, instead of broadcasting the public keys via BLE, the devices only beacon the references (i.e., addresses) of the keys in the bulletin. When a user is infected, a cryptographic hash of encounter tokens is uploaded to the bulletin board. As discussed in Sect. IV-B, TRACECORONA solves this problem by utilizing BLE connections to transfer public keys without any data restrictions.

CleverParrot [15]. To deal with the issue of fitting a DH public key in a BLE advertising message, CleverParrot proposes using a minimum key size of 224 bits (28 bytes) based on the

TABLE IV: The advantages of DH-based approaches in comparison to state-of-the-art approaches. (*) on the user side. (**) Possibly only infected users. (***) prevent one-way and limit real-time two-way attacks. +/- means achieve/not achieve corresponding requirements.

	Centralized	Decentralized		
	BlueTrace/ PEPP-PT/ TousAntiCovid	GAEN/ DP3T-1	DP3T-2/ MIT-PACT/ UW-PACT	TraceCORONA/ Pronto-C2/ CleverParrot
User identifier	Phone number /App ID	Random keys	Random keys	Random keys
Life-time of initial keys	Long-lived	Daily	Short-Lived	Short-lived
Superspreader	+	-*	-*	+*
CAII	+	-*	-*	+*
Identifying users	-	-**	+	+
Tracking users	-	-**	+	+
Extracting social graph	-	-**	+	+
Fake exposure claim	-	-	-	+
Relay attack	-	-	-	+***
Transparency	+	-	+	+
Independency	+	-	+	+

elliptic curve P-224. They choose this key size since it is the same as the one use in Apple's Find My protocol. However, it is worth noting that is a special function in iOS. In fact, both Android and iOS support only 128-bit BLE advertising messages. Therefore, CleverParrot cannot be implemented in practice unless Google and Apple change their BLE platform or they have to adopt and treat CleverParrot as a special function like Apple's Find My.

DH with Private Set Intersection Cardinality (PSI-CA). Epione [17] leverages Function Secret Sharing (FSS) techniques [25] to prevent other users from learning information about the encounter tokens uploaded by infected users. In particular, this approach enables clients (user Apps) in collaboration with the servers *SP* to learn matching encounter tokens, i.e., U_j can know how many encounters with infected users it has without downloading these encounters.

B. Survey on existing DCT schemes, apps and challenges

There are a number of works that survey existing DCT schemes, apps and challenges. Those works can be categorized into two groups: (i) discussing technical specifications, operations and issues of the rolled out apps [26], [27] and (ii) studying certain aspects of some DCT schemes [28], [20]. Sun et al. [26] focus on investigating the security and privacy issues of DCT apps on Android. Wen et al. [27] vet privacy issues of 41 country apps that have rolled our worldwide, in which they focus on analysis of documentation but also binary code to figure out what data an app collects and discuss the potential privacy risks. Unlike those works that focus on the apps, Vaudenay et al. [20] focus on investigating the security and privacy issues of several schemes along with their architectures. The most relevent to our work is the study provided by Ahmed et al. [28]. They discuss 8 different potential attacks on 12 country apps divided in three groups: centralized, decentralized and hybrid architectures. However, those works do not provide an abstraction that groups evaluation requirements of similar schemes as we do in our work.

While existing works point out a number of privacy problems of GAEN [20], [14], [29], [5], Ahmed et al. claim that GAEN protects privacy of users and criticize that existing attacks are unrealistic [30]. However, they do not provide arguments and evidence for their claim, i.e., it is not clear how GAEN can defend against such attacks. In fact, their main experiments only confirm the principal design requirements of GAEN like Randomness of Bluetooth addresses or RPI intervals that are also included in existing attack models [5], [11], [16], [13]. Unfortunately, the paper also gives some misleading information. For example, it states that: "in normal operation, the TEK downloaded are not readily available to the user and the exposure assessment is done away from the user." However, the uploaded TEK keys of infected users are in fact by design public information that is accessible to any moderately sophisticated adversary[7]. For a summary on existing works analyzing DCT, please refer to Tab. VIII of our full technical report [8].

VII. Conclusion

In this work, we propose TraceCORONA that addresses security and privacy challenges of existing contact tracing approaches while providing comparable effectiveness. In contrast to state-of-the-art approaches that are based on exchanging ephemeral IDs, TraceCORONA allows users to anonymously establish encounter-specific tokens using short-range wireless communication like Bluetooth. We systematically and extensively analyze the security and privacy of TraceCORONA in comparison to existing approaches in Sect. V to show that TraceCORONA is resilient to various attacks and thus provides better security and privacy guarantees than other approaches. We have implemented TraceCORONA and published a beta test version of TraceCORONA that has been downloaded and used by more than 2000 users without any major functional problems demonstrating that TraceCORONA is practical. In future work, we will explore approaches to improve the accuracy of distance measuring using ultra-wideband and privacy-enhancing techniques like

[7] An archive collecting *TEK*s of the German DCT App: https://ctt.pfstr.de/

blind signatures to prevent malicious service providers from linking encounter tokens of users.

REFERENCES

[1] L. Ferretti, C. Wymant *et al.*, "Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing," *Science*, 2020.

[2] C. Wymant, L. Ferretti *et al.*, "The epidemiological impact of the nhs covid-19 app." *Nature*, vol. 594, no. 4, 2021. [Online]. Available: https://doi.org/10.1038/s41586-021-03606-z

[3] Apple and Google, "Exposure Notification: Cryptography Specification, v1.2," April 2020, https://www.apple.com/covid19/contacttracing.

[4] Y. Gvili, "Security Analysis of the COVID-19 Contact Tracing Specifications by Apple Inc. and Google Inc." Cryptology ePrint Archive, Report 2020/428, April 2020, https://eprint.iacr.org/2020/428.

[5] L. Baumgärtner, A. Dmitrienko *et al.*, "Mind the gap: Security & privacy risks of contact tracing apps," in *19th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020.

[6] V. Iovino, S. Vaudenay, and M. Vuagnoux, "On the effectiveness of time travel to inject covid-19 alerts," The Cryptographer's Track at the RSA Conference, CT-RSA2021, 2021, https://eprint.iacr.org/2020/1393.

[7] G. Avitabile, D. Friolo, and I. Visconti, "Tenk-u: Terrorist attacks for fake exposure notifications in contact tracing systems," *19th International Conference on Applied Cryptography and Network Security, ACNS2021*, 2021, https://eprint.iacr.org/2020/1150.

[8] T. D. Nguyen, M. Miettinen *et al.*, "Digital contact tracing solutions: Promises, pitfalls and challenges," 2022, available: https://arxiv.org/abs/2202.06698.

[9] J. Bay, J. Kek *et al.*, "Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders," Apr. 2020. [Online]. Available: https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf

[10] D. J. Leith and S. Farrell, "Measurement-based evaluation of google/apple exposure notification api for proximity detection in a light-rail tram," *PLOS ONE*, vol. 15, no. 9, pp. 1–16, 09 2020. [Online]. Available: https://doi.org/10.1371/journal.pone.0239943

[11] C. Troncoso, M. Payer *et al.*, "Decentralized privacy-preserving proximity tracing," *CoRR*, vol. abs/2005.12273, 2020. [Online]. Available: https://arxiv.org/abs/2005.12273

[12] L. Reichert, S. Brack, and B. Scheuermann, "Lighthouses: A warning system for super-spreader events," Cryptology ePrint Archive, Report 2020/1473, 2020, https://eprint.iacr.org/2020/1473.

[13] S. Vaudenay, "Analysis of DP-3T," Cryptology ePrint Archive, April 2020. [Online]. Available: https://eprint.iacr.org/2020/399

[14] L. White and P. van Basshuysen, "Without a trace: Why did corona apps fail?" *Journal of Medical Ethics*, 2021. [Online]. Available: https://jme.bmj.com/content/early/2021/01/08/medethics-2020-107061

[15] R. Canetti, Y. T. Kalai *et al.*, "Privacy-preserving automated exposure notification," Cryptology ePrint Archive, Report 2020/863, 2020, https://eprint.iacr.org/2020/863.

[16] G. Avitabile, V. Botta *et al.*, "Towards defeating mass surveillance and sars-cov-2: The pronto-c2 fully decentralized automatic contact tracing system," CoronaDef Workshop at NDSS 2021, 2021, https://www.ndss-symposium.org/ndss-paper/auto-draft-164/.

[17] N. Trieu, K. Shehata *et al.*, "Epione: Lightweight contact tracing with strong privacy," 2020.

[18] "TraceTogether Contact Tracing App," Government of Singapore, Ministry of Health, 2020, https://www.tracetogether.gov.sg/.

[19] L. Reichert, S. Brack, and B. Scheuermann, "Ovid: Message-based automatic contact tracing," CoronaDef at NDSS 2021, 2021, https://www.ndss-symposium.org/ndss-paper/auto-draft-165/.

[20] S. Vaudenay, "Centralized or decentralized? the contact tracing dilemma," Cryptology ePrint Archive, Report 2020/531, 05 2020, https://eprint.iacr.org/2020/531.

[21] P.-O. Dehaye and J. Reardon, "Swisscovid: a critical analysis of risk assessment by swiss authorities," 2020, https://arxiv.org/abs/2006.10719.

[22] PEPP-PT, "pepp-pt," 2020. [Online]. Available: https://www.pepp-pt.org/content

[23] R. L. Rivest *et al.*, "The pact protocol specification," 2020, https://pact.mit.edu/wp-content/uploads/2020/11/The-PACT-protocol-specification-2020.pdf.

[24] Apple and G. E. N. APIs, https://developers.google.com/android/exposure-notifications/ble-attenuation-overview.

[25] E. Boyle, N. Gilboa, and Y. Ishai, "Function secret sharing," in *Advances in Cryptology - EUROCRYPT 2015*, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 337–367.

[26] R. Sun, W. Wang *et al.*, "An empirical assessment of global covid-19 contact tracing applications," 2021.

[27] H. Wen, Q. Zhao *et al.*, "A study of the privacy of covid-19 contact tracing apps," in *Security and Privacy in Communication Networks*, N. Park, K. Sun *et al.*, Eds. Cham: Springer International Publishing, 2020, pp. 297–317.

[28] N. Ahmed, R. A. Michelin *et al.*, "A survey of covid-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134 577–134 601, 2020.

[29] Z. Brighton-Knight, J. Mussared, and A. Tiu, "Linkability of rolling proximity identifiers in google's implementation of the exposure notification system," Technical report, https://github.com/alwentiu/contact-tracing-research/blob/main/GAEN.pdf.

[30] S. Ahmed, Y. Xiao *et al.*, "Privacy guarantees of ble contact tracing: A case study on covidwise," *arXiv preprint arXiv:2111.08842*, 2021.

BIOGRAPHY

Thien Duc Nguyen is a research assistant at the System Security Lab of Technical University of Darmstadt (TU Darmstadt), Germany. His research interests focus on machine learning-based security mechanisms for IoT, security for federated machine learning, context-based authentication and digital contact tracing.

Markus Miettinen is a postdoctoral researcher at the System Security Lab of TU Darmstadt, Germany. He graduated as Dr.-Ing. from TU Darmstadt in 2018. Before joining academia, he acquired more than a decade of professional experience in industrial research at the Nokia Research Center in Helsinki, Finland and Lausanne, Switzerland. His research interests lie in utilizing machine learning methods for realizing autonomous security solutions for IoT and mobile computing environments.

Alexandra Dmitrienko is an associate professor and the head of the Secure Software Systems group at the University of Wuerzburg in Germany. She holds a PhD degree in Security and Information Technology from TU Darmstadt (2015). She received numerous awards for her research, including Intel Doctoral Student Honor Award (2013) and ERCIM STM WG Award (2016). Her research interests focus on secure software engineering, systems security and privacy, and security and privacy of mobile, cyber-physical, and distributed systems.

Ahmad-Reza Sadeghi is a full professor for Computer Science at TU Darmstadt, where he directs the System Security Lab, the Intel Private AI Center, and Open S3 Lab. He received his PhD from the University of Saarland. He has served on the editorial board of ACM TISSEC and as Editor-in-Chief for IEEE Security and Privacy Magazine.

Ivan Visconti is a full professor of Computer Science in the Computer and Electrical Engineering and Applied Mathematics Department of the University of Salerno. His research interests focus mainly on designing provably secure and privacy-preserving cryptographic protocols and securing blockchains and their applications. He served for two years as Senior Area Editor for the journal IEEE Transactions on Information Forensics and Security.