DOI:10.1145/3524107

Lessons from a pandemic.

Deploying Decentralized, Privacy-Preserving Proximity Tracing

CARMELA TRONCOSO
EPFL, SWITZERLAND

DAN BOGDANOV
CYBERNETICA AS, ESTONIA

EDOUARD BUGNION
EPFL, SWITZERLAND

SYLVAIN CHATEL
EPFL, SWITZERLAND

CAS CREMERS
CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY, GERMANY

SEDA GÜRSES
DELFT UNIVERSITY OF TECHNOLOGY, NETHERLANDS

JEAN-PIERRE HUBAUX
EPFL, SWITZERLAND

DENNIS JACKSON
MOZILLA, U.K.

JAMES R. LARUS
EPFL, SWITZERLAND

WOUTER LUEKS
EPFL, SWITZERLAND

RUI OLIVEIRA
UNIVERSITY OF MINHO AND INESC TEC, PORTUGAL

MATHIAS PAYER
EPFL, SWITZERLAND

BART PRENEEL
KU LEUVEN AND IMEC, BELGIUM

APOSTOLOS PYRGELIS
EPFL, SWITZERLAND

MARCEL SALATHÉ
EPFL, SWITZERLAND

THERESA STADLER
EPFL, SWITZERLAND

MICHAEL VEALE
UNIVERSITY COLLEGE OF LONDON, U.K.

CONTACT TRACING IS a time-proven technique for breaking infection chains in epidemics. Public health officials interview those who come in contact with an infectious agent, such as a virus, to identify exposed, potentially infected people. These contacts are notified that they are at risk and should take efforts to avoid infecting others—for example, by going into quarantine, taking a test, wearing a mask continuously, or taking other precautionary measures.

In March 2020, as the first wave of the COVID-19 pandemic was peaking, traditional manual contact tracing efforts in many countries were overwhelmed by the sheer volume of cases; by the rapid speed at which SARS-CoV-2 spread; and by the large fraction of asymptomatic, yet infectious, individuals.

PHOTO BY BANGKOK CLICK STUDIO

Many people quickly and independently proposed using ubiquitous smartphones to implement *digital contact tracing* (*DCT*). In this new approach, an app on a user's phone could record contacts (encounters with other people) of sufficient time duration. If a physically close contact was diagnosed as infected, the app could inform the phone's potentially infected user. The envisioned technology would complement manual contact tracing by notifying people faster; reducing the burden on trained contract tracers; increasing scalability; and finding anonymous contacts, such as those in public spaces like shops and transportation, who would be otherwise unreachable through traditional systems.

Due to the fast-moving pandemic, the need for DCT was urgent, and had to be designed, developed, and deployed in a highly compressed timeline. This pressure limited the design scope and constrained many decisions. For example, manufacturing and distributing new hardware to the public would have incurred substantial delays, so viable solutions could only make use of sensor technology already widely deployed on consumer mobile phones and existing communication infrastructure.

A further challenge was to ensure the infrastructural components deployed for DCT could not be used to invade individual privacy or facilitate human rights abuses. For example, DCT applications that collect and share timestamped and geo-located records of people's physical contacts can be easily repurposed for illegitimate, oppressive uses beyond public health. This happened with contact-tracing information collected in paper form[19,39] and has led to increased surveillance[2] and stigmatization.[25] Moreover, databases recording peoples' locations are susceptible to being leaked, intentionally or unintentionally.[34] During an event requiring an internationally coordinated response, the potential for abuse of a new technology could not be ignored at the design stage, especially with respect to the varying political and governmental systems and rule of law (particularly during states of emergency).

Furthermore, trustworthy and transparent technology is essential to achieve the high voluntary adoption necessary for maximal public health impact, particularly in countries with a history of poor data management. Achieving and retaining trust is an international effort—data breaches or misuse in one country can resonate around the world. Thus, limiting abuse is fundamental to achieving public health goals.

» key insights

- **It is possible to build privacy-preserving systems that not only collect and process little information but ensure information can only be used for a single purpose. Our contact-tracing system can only be used to notify contacts.**
- **Successful deployment of privacy-preserving solutions requires consideration of the broader context in which these solutions must operate. For example, integrating contact-tracing apps with public health systems is essential.**
- **Reliance on third-party technologies, in particular mobile platforms, severely constrains the deployment of privacy-preserving systems.**

The authors represent a major portion of the group that designed the Decentralized Privacy-Preserving Proximity Tracing (DP-3T) protocol, which heavily influenced the Google and Apple Exposure Notification (GAEN) framework used by most DCT apps, and we helped deploy five apps: SwissCOVID[a] (Switzerland), Corona Warn App[b] (Germany), STAYAWAY COVID[c] (Portugal), Coronalert[d] (Belgium), HOIA[e] (Estonia), and the European Federated Gateway Server.[f] In this article, we describe the lessons learned from our efforts to design, develop, and deploy digital contact tracing. We detail the hurdles and challenges, including the design of underlying cryptographic protocols, the development of mechanisms to ensure end-to-end privacy for users, and the integration of the apps within public health systems. We also discuss some issues raised in the media concerning the contact-tracing apps' effectiveness, security, and independence from device manufacturers. We conclude with recommendations to help prepare technologically for the next emergency.

Digital Privacy-Preserving, Proximity-Tracing Protocols

In response to a pressing need for DCT to combat the COVID-19 pandemic, a substantial number of proposals were put forward, using a diverse range of technologies including Bluetooth, Ultrasound, and the Global Navigation Satellite System (GNSS). Of these, Bluetooth was most widely adopted, in large part due to privacy concerns around access to microphone and location information. In this article, we focus exclusively on proposals that use Bluetooth Low Energy (BLE) beacons to detect proximity without knowledge of a contact's location or identity.

Common to these proposals, a smartphone runs a contact-tracing app that executes a Bluetooth contact-tracing protocol, also known as a proximity-tracing protocol. As such, devices broadcast ephemeral identifiers using BLE beacons. Broadcasting, instead of point-to-point connections, ensures that the number of devices able to receive the identifiers is not limited by a phone's Bluetooth connection rate. A phone that receives such an identifier records it, alongside information about the signal's power, which can be used to estimate the proximity of the transmitting device. Later, these records provide a basis for a risk calculation based on the estimated proximity to contagious individuals.

The proposals differ on how the risk calculation is carried out, and on the capabilities they require from mobile phones and the communication infrastructure. A few dozen non-deployed academic proposals explore different privacy/security trade-offs—for example by using more complex cryptography[7,9,37] or requiring currently nonexistent or non-scalable infrastructure as a basis for their strong privacy guarantees.[1,3,9]

We only discuss the BLE-based protocols whose design allows immediate deployment. We categorize them as either centralized or decentralized according to their risk calculation. In the former, a central server carries out the risk calculation on behalf of all users and notifies those it considers to be at risk. In the latter, each user's device carries out an individual risk calculation to

[a] https://foph-coronavirus.ch/swisscovid-app/
[b] https://www.coronawarn.app/en/
[c] https://stayawaycovid.pt/landing-page/
[d] https://coronalert.be/en/
[e] https://hoia.me/en/
[f] https://ec.europa.eu/commission/presscorner/detail/en/ip_20_1904

decide whether to notify the user. For both, we provide a high-level description of their operation and their security and privacy properties. For more, we refer the reader to our detailed analysis and comparison.[14]

Centralized proximity-tracing protocols. This class of contact-tracing protocols,[4,13,28,32] pioneered by Singapore's BlueTrace app,[4] uses a central server to generate ephemeral identifiers that a phone downloads and periodically broadcasts. When a user receives a positive COVID-19 diagnosis, that user's phone app uploads all the identifiers it received to the server. The server performs a matching process on these identifiers to determine who was in prolonged contact with the COVID-19-positive person and notifies those people of a potential exposure.

Due to the fast-moving pandemic, the need for DCT was urgent, and had to be designed, developed, and deployed in a highly compressed timeline.

Security and privacy. In a *centralized* design, the server generates the ephemeral identifiers and associates them with the long-term identities that are necessary to send notifications. Therefore, an adversary with access to the server can de-anonymize any observed BLE beacons. Moreover, an adversary who can influence the server can falsely notify a user of infection. In addition, the server's information allows inference of relationships among users (who they met, when, and for how long). This information can be inferred even for users who do not test positive, so long as they come in contact with a positive user. This can lead to scope creep, where the system is explicitly or implicitly repurposed, such as in Singapore, where the police were given access to the app-related databases for law enforcement purposes.[23]

Decentralized proximity-tracing protocols. To avoid the security and privacy shortcomings of the centralized approach, a number of *decentralized* protocols[8,10,31,36,38] moved the generation of the ephemeral identifiers broadcast in BLE beacons and the matching process to run entirely on an individual's smartphone. This design reduces the power of the central server by limiting its role to checking that a user has been diagnosed by a healthcare provider and to distributing public information.

We now present our design, the DP-3T protocol.[38] Other protocols, which appeared concurrently, are very similar.[10,13,36] In a setup phase, a phone creates a secret seed SK_t. After that, the DP-3T protocol operates in three steps:

1. **Local ephemeral identifier creation.** The phone app creates its ephemeral identifiers (EphIDs) using the following procedure:

- Each day, the secret seed is rotated using a simple, non-reversible transformation: $SK_{t+1} = H(SK_t)$, where H is a hash function.
- Phones derive $n = (24 * 60)/L$ ephemeral identifiers (EphIDs) from the daily seed: $\texttt{EphID}_1||\ldots||\texttt{EphID}_n$ = PRG(PRF(SK_t, "broadcast key")), where PRF is a pseudo-random function (for example, HMAC-SHA256), "broadcast key" is a fixed public string, and PRG is a pseudo-random generator (for instance, AES in counter mode).

Each EphID is broadcast for L minutes. The value L is public. Smartphones pick a random order in which to broadcast these EphIDs. One cannot link two such identifiers without knowing the key SK_t.

2. **Operation: Storage of beacons and seeds.** For each received beacon, a phone stores:

- The received ephemeral Bluetooth identifier EphID.
- The exposure measurement (for example, signal attenuation).
- The day on which this beacon was received (for instance, "April 2").

Phones store beacons indexed by EphID. In addition, each device stores the seeds SK_t it generated for as long as recommended by health authorities (for example, 14 days).

3. **Local notification procedure.** Users who receive a positive COVID-19 test are authorized by the health authority to upload the seed SK_t to the central server, corresponding to the first day when they are likely to have been contagious. After the upload, user devices randomly generate a new secret seed SK_t to prevent linkability with respect to previous secret keys.

All phones periodically download the seeds of COVID-19-positive users from this server. With each seed, a smartphone can locally reconstruct the list of EphIDs broadcast by a diagnosed person for one day. The app matches these EphIDs to check two things: If the phone observed a beacon with one of these EphIDs and if the observation occurred before the corresponding seed

SK_t was published (to avoid replay attacks in which EphIDs are re-derived from a published SK_t and retransmitted). Using the signal strength of the set of observed beacons, the smartphone locally computes how long and at what distance its user was exposed to COVID-19-positive people. If the time frame is long and the distance close enough, the phone notifies its user of a high-risk contact indicating a possible contagion.

Security and privacy. In this decentralized design, the server has no information to link ephemeral identifiers. The server also cannot influence the generation of identifiers or arbitrarily mark users as at-risk. Most implementations of these protocols try to reduce the amount of information transmitted by the server to save bandwidth. Unfortunately, more bandwidth-efficient implementations (such as the one described earlier) enable linking of beacons broadcast by positive users on the days they were infectious. The protocol fully protects non-positive users. Slight changes in the cryptographic protocols can combat linkability, at the expense of increasing bandwidth (see unlinkable scheme in Troncoso et al.[38]).

From DP3T to GAEN. Google and Apple subsequently implemented a decentralized DCT framework, very similar to DP-3T,[38] in their GAEN framework.[18] The main difference is the creation of a fresh new key every day and the use of a different derivation to create the EphIDs.

This framework is currently the basis of more than 40 DCT apps in Europe and North and South America; the number of downloads is estimated to be at least 90 million. Almost all European countries and U.S. states adopted the decentralized approach because of its strong privacy benefits and support from mobile operating-system vendors. Currently, apps from 14 countries in the EU are connected through the European Federated Gateway System.[17] This gateway enables exchanges between apps using the GAEN decentralized framework (see section titled Integration Across Health Systems). National applications were used throughout the pandemic, as part of national test-and-trace strategies. In most European countries, DCT apps were suspended together with test-and-trace during the first half of 2022.

Trustworthy and transparent technology is essential to achieve the high voluntary adoption necessary for maximal public health impact.

From Protocol to System: Integration Challenges

While instrumental to the operation and the security and privacy guarantees of a DCT app, a proximity-tracing protocol is just a small piece in the larger challenge of reducing COVID-19 transmission. This protocol must be implemented in mobile apps that run on a large and diverse collection of phones that differ in firmware and hardware. In turn, the apps and server must be integrated into a public health system that acts as an interface between health services and users. Each step of integration brings new operational challenges and difficulties in maintaining end-to-end security and privacy.

Integration with existing hardware. Bluetooth's ubiquity offers a solid basis for building widely deployed privacy-preserving systems. However, Bluetooth also imposes numerous constraints. For example, support in hardware and operating systems varies widely, often exposing differing APIs with limited functionality to an application. Apple's *CoreLocation* API allows BLE to function much more extensively in the background than its generic *CoreBluetooth* API yet functions only with proprietary "iBeacons," thereby prohibiting interaction with non-Apple devices. Similarly, capabilities concerning transmission power, tag options, or permissions vary between Android versions. A further complication is ensuring a system has minimal impact on battery life while maintaining reliable message reception and transmission.

Many proposals opted for a connectionless broadcast system that requires each phone to produce a constant stream of broadcast messages. While a connection-based approach could in theory be used, it would require substantially more battery power to maintain a connection with each nearby phone and would encounter interference problems in crowded environments. Moreover, the most widely used Bluetooth broadcast standard supports only relatively small beacon payloads, which imposes another design constraint. Another privacy problem is the highly varied support for Bluetooth privacy extensions, such as rotating MAC addresses. Although address rotation is recommended to mitigate user tracking, many devices do not support it. Without it, an adversary can track users (as the MAC address remains the same with an effect comparable to broadcasting a single static ID), despite the cryptographic precautions included in DCT proposals.[14]

Integration in the mobile operating system. When designing DCT protocols, designers assume these protocols will be part of a DCT app and will operate independently of the mobile operating system. However, the highly integrated design of mobile-phone platforms means that a DCT protocol must be integrated into a phone's operating system. This is necessary to guarantee that beacons will be reliably sent and received, to limit battery consumption, and to ensure that protection at the hardware level (for example, MAC rotation) is applied. Next, we describe how the consequences of this integration strongly impact the way apps operate and can be deployed.

The GAEN API[18] went beyond the necessary integration. It presented an app with an API with a heavily constrained set of parameters. These constraints strongly limited the design choices of app developers in making tradeoffs among privacy, security, and epidemiological utility of the applications.

For example, the first version of the GAEN API provided apps with only heavily summarized information about observed beacons, with the operating system performing the exposure computation and providing a result through API calls. Initial API versions allowed only limited forms of aggregation. Specifically, it did not permit computation of daily viral exposure accumulation. As a result, early versions of SwissCovid, Radar COVID (Spain), STAYAWAY COVID, and HOIA, whose epidemiology experts opted for day-based computation, had to work around the limitations by performing multiple API queries, thereby delaying notifications for up to eight hours. This was eventually resolved in GAEN version 1.6.

As a further example, the early version of the GAEN API did not permit the release of daily keys until they expired. This security mechanism, aimed at reducing the likelihood of a replay attack, affected how apps could upload keys to the central server. An authorized user could still be infectious; thus, that user should upload keys up to and including the day of upload. As a result, most apps had to change their original authorization schemes to perform a second authorization to upload on the subsequent day without user intervention. This not only changed the app's functional flows, but also changed the security analysis: A second upload behind a user's back carries the risk that its authorization can be misused to upload unauthorized cryptographic material to the server.

Finally, how and when the GAEN API was integrated in the operating system strongly affected the availability of DCT apps. For example, old versions of iOS, such as those for the iPhone 6, were only supported six months after the initial release of the framework. This affected a non-negligible number of users, who lost interest in using the app. Other older iOS versions, even those originally supported by the framework, do not have good background task management, which hinders the apps by not permitting them to wake up periodically to download new information about infected users.

Integration into a health system. In all DCT applications, a key step in the process is when a COVID-19-positive person uploads ephemeral identifier seeds to a server. To ensure that only infected users upload their identifier seeds, minimizing the likelihood of fake alerts, apps require an authorization key from the health system. However, only a small number of proposals specified how this key could be securely provided.[9,15]

While this process may seem simple, it has proven to be a major challenge because most countries' health systems lack a comprehensive digitized framework to manage aspects of the pandemic response, including test results and interactions with people. Often these systems are not even computerized and consist of a few disconnected databases and personnel who cannot communicate digitally with patients. To deal with this situation, most apps use a very simple authorization mechanism that is communicated to users by phone or SMS. Moreover, the DCT infrastructure needed to be developed, maintained, and secured on an ongoing basis, often by governmental departments that lack the experience or competence to directly run such services.

Integration across health systems. Finally, the pandemic is a global problem. Around the world, people frequently travel and commute between

PHOTO BY PROSTOCK-STUDIO

states or countries. In such situations, apps must be able to trigger notifications across borders.

When designing DCT protocols, ease of interoperability was a consideration. Exchange of information across borders is in principle facilitated by the privacy guarantees of the decentralized protocols. In these protocols, only the keys from infected users must be exchanged. The keys are not sensitive since they carry no information about individuals, their location, or their interactions with others.

In practice, legal experts have categorized the uploaded seeds as pseudonymous personal data under GDPR, which means that legal rules influence how they can be shared and with whom. Such legal considerations hindered the fast deployment of the European Federated Gateway Server (EFGS)[17] used by the decentralized apps in most countries in Europe to exchange keys.

Interoperability also becomes complex when countries configure the GAEN API in different ways to estimate exposure risk. Even if a country uses a simple set of the parameters for its own risk function—for example, SwissCovid[35]—its server may need to collect extra information to support the more complex risk functions of other countries—for instance, CoronaWarnApp.[11] This necessitates the creation of common standards to exchange metadata associated with keys and complicates the logic that interprets and supports other risk functions.

Increasing complexity can also affect the privacy promised from a country's app to its citizens. As extra information is published by other countries to enable their risk estimation, this information becomes available to an adversary, who can use it to reduce the anonymity sets of users. Mitigating this leakage requires developers to carefully select the information that is shared to minimize inferences while still enabling meaningful risk estimation.

Deploying Large-Scale DCT: Lessons Learned

The challenges noted in the previous section hindered the deployment of these apps at many steps, requiring extra engineering to build and deploy the apps at a large scale. During deployment in several countries, we learned many valuable lessons which highlight how often research and academic work are not aligned with real-world demands. Academic research sometimes aims toward a level of perfection beyond that which is demanded in real situations. Academic work, moreover, typically focuses on one aspect of a system and requires additional mechanisms to ensure that the security and privacy properties encompass all the elements of a deployed system, from phone to cloud.

Privacy must be guaranteed at all layers. Long discussions in academia and public forums focused on competing DCT protocols and their security and privacy properties. However, as we previously noted, the cryptographic protocol is just a small part of a DCT system. Information flows that complement the low-level protocol can result in privacy leaks that must be prevented with additional mechanisms.

Privacy at the network layer. Information uploaded to the server cannot be linked to the identity of users reporting their positive status. However, the mere existence of the connection to upload this information reveals the health status (SARS-CoV-2-positive) of the user to any adversary who can see a user's IP address (for example, an eavesdropper on a Wi-Fi network or the Internet service provider).

In academia, the typical mitigation is for the app to generate dummy traffic, in addition to its real traffic, to help obscure when real actions occur. Even though well-known and frequently proposed, dummy traffic is non-trivial to properly implement. For instance, configuring traffic requires an understanding of the actual usage patterns that must be mimicked. In reality, this pattern is often unknown, particularly for a new and unprecedented service such as DCT. One simple option is to over-provision the dummy traffic, but this could affect the user experience by reducing battery life and consuming data bandwidth. To balance these requirements, instead of aiming to make dummy and real traffic indistinguishable, plausible deniability is a more attainable goal. This enables the system to deny that some actions are real (in this case to deny that actual uploads happened by claiming uploads are dummies). It is typically considered weak in an academic publication but often suffices in practice.

Privacy of the authentication scheme. Besides hiding the traffic patterns associated with an upload, it is also important that the authentication mechanism does not provide additional information about (1) the identity of the user or (2) the link between a user and the information the user uploads.[15]

While powerful cryptographic tools exist to achieve security and unlinkability at the same time, such as anonymous credentials or cryptographic commitments, using them in practice requires a highly digitalized health system not available in most countries. Further, even with systems using anonymous credentials, many orthogonal means of inference are available to an adversary, such as timing or IP-address metadata. As this latter class of metadata is very difficult to conceal,[g] most apps use a simple code-based authentication scheme and trust the servers to not log information that would enable the linkage of users' IPs and their ephemeral identifiers.

Contact-tracing applications are most valuable when widely used, so some countries opted to host their servers in public clouds to support high loads. Other countries hosted their servers locally in infrastructure owned by the government or local companies. Whether hosting is public or private, a large number of users requires technologies, such as load balancers and firewalls, that can log information outside of the control of the app designers. Careful design of the app server's logging policy is vital to ensure that none of the information logged by the cloud infrastructure can be used to breach the app user's privacy.

Exposure estimation goes beyond distance measurement. Another point of contention in academic circles and public discussions is the accuracy of BLE when measuring distance and its suitability as the underlying technology for DCT.[27,41] In practice, while accuracy matters and improvements in distance measurement at the Bluetooth layer would be valuable,[22,29] it is important to remember that the goal

[g] Contemporary metadata hiding systems such as Tor do not scale to hundreds of millions of users.

of a contact-tracing app is not to measure a precise distance at one point in time but instead to estimate a person's exposure to other COVID-19-positive people over a period of time.

It is also important to keep in mind that the epidemiological basis for computing exposure is not an exact science. The technological solution mimics a contact-tracing interview in which patients are asked to recall close contacts, typically defined as those occurring longer than 15 min. within 2 m (6 ft.). In such a situation, a patient's estimation of distance is naturally limited in precision and accuracy by human perception and memory. Moreover, the 2-m criterion is itself an approximation. There is no specific distance at which the virus stops traveling, and the high-risk zone depends very heavily on environmental conditions, such as air circulation. Later in the article, we discuss complementary protocols for notifying about contamination in poorly ventilated spaces in which contagion can occur well beyond 2 m.

Third, the computation of exposure with the GAEN framework is constrained by the frequency of measurement and the information exposed to apps via the framework's API, which limits how information can be combined. As a result, existing contact-tracing apps follow diverse strategies, ranging from very simple approaches[35] to complex formulae.[6,11] These constraints on the exposure computation also reduce the importance of distance measurement accuracy, as it gets diluted in the aggregation function and degraded by the measurement frequency.

Even the best technology underperforms if not used. Researchers and developers have focused on optimizing the technology by improving measurement accuracy and proposing many variations to the protocol. These alternatives offer different tradeoffs among security, privacy, and device capabilities (for example, battery consumption, sensor usage, and use of devices beyond the phone). However, in the end, no improvement can increase the value of a technology that lacks broad adoption. Achieving this end requires good integration, not only in a technical sense but also with the processes and individuals in the broad environment.

Google and Apple subsequently implemented a decentralized DCT framework, very similar to DP-3T, in their Exposure Notification framework.

The environment of contact-tracing apps is complex, with many stakeholders: governments, public health services and employees, mobile operators, mobile operating systems developers, and, of course, users. In this article, we have discussed the technical integration difficulties arising from the strong dependence of these apps on the operating system and changes by mobile-system developers. However, throughout the world, the principal difficulties confronting these apps arise in procedural and social integration:

Rollout by public entities. Rollout of this technology by health authorities in numerous countries was a complicated process involving numerous facets of the public health system as well as the unfamiliar deployment of technical infrastructure and a publicly available app. To start, each health authority needed to procure or otherwise build a DCT system for its local market. Open source, such as that produced by DP-3T and other projects, helped development in countries with fewer resources. In addition, Apple and Google support development efforts in many countries and incorporated an Express app into their later OS releases. Even with this support, not all countries could promptly roll out an application-supported DCT system or maintain it properly.

External dependencies. The operation cycle of contact-tracing apps is not completely technical.[5] Two crucial steps are outside the protocol: delivering the authorization code to users and contacting the health system after a notification. In most countries, these processes require human intervention and have proven to be the least reliable part of the system. The difficulty of incorporating these steps into existing medical practice introduces delays and communication failures that decrease the health impact and may eventually cause users to abandon the application, leading to a public perception that the app is not useful or functional.

Bottlenecks also appeared in systems that did not use authorization codes and instead integrated with national e-health records holding test results. In Estonia, requiring strong authentication for infection confirmation proved to be the limiting factor. Even though multiple authentication methods were offered, not all patients with positive COVID-19 test results had access to at least one method for confirming their infection, preventing them from notifying others.

Communication strategy. The adoption of contact-tracing apps depends on multiple factors.[21,24] Among them is a user's perception of the utility and risks stemming from using an app. Both studies, and results in practice, show that adoption is greatly hindered by doubts about the app's accuracy due to its use of Bluetooth, a technology not designed for distance measurements, as well as concerns about its privacy properties.

The importance of privacy concerns came as a surprise considering our efforts to minimize the data used by the applications. Unfortunately, the privacy properties of these apps are understood only by experts. Moreover, the early public and heated debate about the advantages and disadvantages of centralized and decentralized apps may have exacerbated this confusion. Most users have no means to verify an app and find it difficult to believe that it will not collect data (given that this is not true for almost any other app on their phone). Digital literacy is increasingly essential to enable non-experts to actively participate in this type of public discussion.

Concerns about the accuracy and effectiveness of the app are complex to explain. Moreover, the strong privacy protection of the apps does not permit immediate collection of statistics to demonstrate its value. It is possible, however, to gather data about DCT outside of the app.[5] Using these other means, researchers have demonstrated the effectiveness of the apps in at least three countries.[12,16,33,40] Among others, these studies show that apps have similar second-attack rates to manual tracing, provide faster notifications than manual contact tracing for users who do not live in the same household, and reach wider circles than contacts manually reported by index cases. These findings, which could build confidence in the value and effectiveness of these apps, are also difficult to communicate clearly and understandably to the general population. In some countries, communications so far have mainly been confined to researchers and not authorities, limiting its value in increasing DCT usage.

The highly integrated design of mobile-phone platforms means that a DCT protocol must be integrated into a phone's operating system.

Looking Ahead: Paving the Way to Respectful Technology for the Next Emergency

The DP3T project is proof that it is possible to build and deploy practical, scalable, and useful privacy-preserving applications without collecting data that could be abused. At the same time, our deployment experience demonstrated the difficulty of achieving a high level of end-to-end privacy. Delivering a high assurance in the design required months of iterative effort to overcome the many practical obstacles to privacy that have their roots in today's service-oriented software engineering practices.[26]

A large part of this difficulty stemmed from our reliance on the smartphone ecosystem, which is tightly controlled by Apple and Google. The involvement of these two giants came with notable advantages: It quickly established a de facto international standard and rapidly brought together resources from the two companies to build and deploy efficient GAEN implementations. However, their involvement meant that these two companies decided which DCT applications were permitted and how they could operate.

Two changes could make future public health deployments faster and less dependent on big tech. On the public sector side, there is a pressing need for improved independent infrastructure and software development capability. On the platform side, it is imperative that mobile application development patterns emerge that are architecturally separate from the core operating system provided by Google and Apple, so that the key control points (app delivery, update, notification, and more) are not solely under the control of operating system providers.[20] This does not mean removing all control points entirely, which might lead to security vulnerabilities. But having regard for security in software development and distribution does not necessarily entail giving a small number of firms the magnitude of decision-making power we currently do. It is important for the research community, and for society, that alternatives to these proprietary platforms emerge so that future public health software can be effective, fully accountable, and auditable. The need to rethink the current landscape is increasingly subject to political attention, such as through third-party app store provisions in the EU's *Digital Markets Act*.

Despite the strong protections embedded in decentralized DCT protocols, the limited adoption of these applications in some countries hampered their efficacy. However, other countries saw reasonably high adoption levels (U.K., Finland, Netherlands, Ireland, and Germany), and there is evidence that the apps warned millions of users, in many cases faster than manual contact tracing.[33,40] In the future, it is important to increase adoption by accelerating the collection of evidence of effectiveness with integrated, privacy-preserving metrics from the onset. However, mechanisms to compute such metrics are hard to integrate in practice.[26]

The design principles behind DCT apps can be harnessed to build other applications to help with pandemic containment. For instance, there is growing evidence that SARS-CoV-2 can be transmitted beyond close-proximity contacts. Decentralized technologies can also be used to efficiently notify visitors of venues and events with SARS-CoV-2-positive attendees.[30]

The same design philosophy, centered on limiting the purpose of applications, can be applied to other new technologies, especially if their effects are uncertain as in the case of pandemic-mitigation technical solutions. By following this path, we can harness the potential benefits of technology, without endangering the fundamental societal values of liberty, freedom, and the right to privacy.

Acknowledgments

The authors would like to thank all the individual researchers, companies, organizations, and public health officials that participated in the deployment and evaluation of COVID-19 DCT apps. This project has been partially funded by Fondation Botnar, Basel, Switzerland. **Carmela Troncoso** (camela.troncoso@epfl.ch) served as contact author for this article.

References

1. Avitabile, G., Botta, V., Iovino, V., and Visconti, I. Towards defeating mass surveillance and SARS-CoV-2: The Pronto-C2 fully decentralized automatic contact tracing system. *IACR Cryptology ePrint Archive* (2020), 493. https://eprint.iacr.org/2020/493.
2. Bahrain, Kuwait and Norway contact tracing apps among most dangerous for privacy. Amnesty International (2020), https://www.amnesty.org/en/ latest/news/2020/06/bahrain-kuwait-norway-contact-tracing-apps-danger-for-privacy/.
3. Barthe, G. et al. PanCast: Listening to Bluetooth beacons for epidemic risk mitigation. *CoRR* abs/2011.08069 (November 2020), https://arxiv.org/abs/2011.08069.
4. Bay, J. et al. BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders. (2020), https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf.
5. Benzler, J. et al. Towards a common performance and effectiveness terminology for digital proximity tracing applications. *CoRR* abs/2012.12927 (December 2020), https://arxiv.org/abs/2012.12927.
6. Briers, M., Charalambides, M., and Holmes, C. Risk scoring calculation for the current NHSx contact tracing app. *CoRR* abs/2005.11057 (May 2020), https://arxiv.org/abs/2005.11057.
7. Canetti, R. et al. Privacy-preserving automated exposure notification. *IACR Cryptology ePrint Archive* (2020).
8. Canetti, R., Trachtenberg, A., and Varia, M. Anonymous collocation discovery: Harnessing privacy to tame the coronavirus (March 2020), https://arxiv.org/abs/2003.13670.
9. Castelluccia, C. et al. DESIRE: A third way for a European exposure notification system leveraging the best of centralized and decentralized systems. *CoRR* abs/2008.01621 (August 2020), https://arxiv.org/abs/2008.01621.
10. Chan, J. et al. PACT: Privacy sensitive protocols and mechanisms for mobile contact tracing. *CoRR* abs/2004.03544 (April 2020), https://arxiv.org/abs/2004.03544.
11. CWA Team. Epidemiological motivation of the transmission risk level. (October 2020), https://github.com/corona-warn-app/cwa-documentation/blob/main/transmission_risk.pdf.
12. Daniore, P., Ballouz, T., Menges, D., and von Wyl, V. The SwissCovid digital proximity tracing app after one year: Were expectations fulfilled? *Swiss Medical Weekly* (September 2021).
13. Data protection and information security architecture. PEPP-PT (Apr. 2020), https://github.com/pepp-pt/pepp-pt-documentation/blob/master/10-data-protection/PEPP-PT-data-protection-information-security-architecture-Germany.pdf.
14. DP-3T Team. Privacy and security risk evaluation of digital proximity tracing systems (April 2020), https://github.com/DP-3T/documents/blob/master/Security%20analysis/Privacy%20and%20Security%20Attacks%20on%20Digital%20Proximity%20Tracing%20Systems.pdf.
15. DP-3T Team. Secure upload authorisation for digital proximity tracing (April 2020), https://github.com/DP-3T/documents/blob/master/DP3T%20-%20Upload%20Authorisation% 20Analysis%20and%20Guidelines.pdf.
16. Ebbers, W., Hooft, L., van der Laan, N., and Metting, E. Evaluation CoronaMelder: An overview after 9 months. https://www.coronamelder.nl/media/Evaluatie_CoronaMelder_na_9_maanden_english.pdf.
17. European Interoperability Certificate Governance. A security architecture for contact tracing and warning apps. *eHealth Network* (April 2020), https://health.ec.europa.eu/publications/european-interoperability-certificate-governance-security-architecture-contact-tracing-and-warning_en.
18. Exposure notification—Cryptography specification. Google LLC and Apple Inc. (April 2020), https://blog.google/documents/69/Exposure_Notification_-_Cryptography_Specification_v1.2.1.pdf.
19. German restaurants object after police use COVID data for crime-fighting. *Reuters* (July 2020), https://www.reuters.com/article/us-health-coronavirus-germany-privacy-idUSKCN24W2K6.
20. Groschupp, F. et al. Sovereign smartphone: To enjoy freedom we have to control our phones. arXiv:2102.02743 (Feb. 2021).
21. Hargittai, E., Redmiles, E.M., Vitak, J., and Zimmer, M. Americans' willingness to adopt a COVID-19 tracking app. *First Monday 25*, 11 (Oct. 2020).
22. Hatke, G.F., et al. Using Bluetooth Low Energy (BLE) signal strength estimation to facilitate contact tracing for COVID-19. *Pact Technical Report* (2020).
23. Illmer, A. Singapore reveals Covid privacy data available to police. *BBC* (Jan. 2021), https://www.bbc.com/news/world-asia-55541001.
24. Kaptchuk, G. et al. How good is good enough for COVID19 apps? The influence of benefits, accuracy, and privacy on willingness to adopt. arXiv:2005.04343 (May 2020).
25. Kim, N. 'More scary than coronavirus': South Korea's health alerts expose private lives. *The Guardian* (March 2020), https://www.theguardian.com/world/2020/mar/06/more-scary-than-coronavirus-south-koreas-health-alerts-expose-private-lives.
26. Kostova, B., Gürses, S., and Troncoso, C. Privacy engineering meets software engineering. On the challenges of engineering Privacy By Design. (July 2020), arXiv:2007.08613.
27. Leith, D.J. and Farrell, S. Coronavirus contact tracing: Evaluating the potential of using Bluetooth received signal strength for proximity detection. *Computer Communication Review 50*, 4 (October 2020), 66-74.
28. Levy, I. High level privacy and security design for NHS COVID-19 contact tracing app. National Cyber Security Centre, U.K. (May 2020), https://www.ncsc.gov.uk/files/NHS-app-security-paper%20V0.1.pdf.
29. Lovett, T. et al. Inferring proximity from Bluetooth Low Energy RSSI with unscented Kalman smoothers. *CoRR* abs/2007.05057 (July 2020), https://arxiv.org/abs/2007.05057.
30. Lueks, W. et al. CrowdNotifier: Decentralized privacy-preserving presence tracing. In *Proceedings on Privacy Enhancing Techniques 2021*, 4 (July 2021), 350-368.
31. Rivest, R.L. et al. The PACT Protocol technical specification. PACT (April 2020), https://pact.mit.edu/wp-content/uploads/2020/11/The-PACT-protocol-specification-2020.pdf.
32. ROBERT: ROBust and privacy-presERving proximity Tracing. Inria PRIVATICS team and Fraunhofer AISEC (April 2020), https://github.com/ROBERT-proximity-tracing/documents/blob/master/previous_versions/ROBERT-specification-EN-v1_0.pdf.
33. Salathé, M. et al. Early evidence of effectiveness of digital contact tracing for SARS-CoV-2 in Switzerland. *Swiss Medical Weekly 150*, 2020/5153 (December 2020).
34. Sterling, T. Personal data stolen from Dutch coronavirus track-and-trace programme. *Reuters* (2021), https://www.reuters.com/article/us-health-coronavirus-netherlands-datapr-idUSKBN29Y1H3.
35. SwissCovid Exposure Score Calculation. https://github.com/admin-ch/PT-System-Documents/blob/master/SwissCovid-ExposureScore.pdf.
36. TCN Coalition. TCN Protocol (2020), https://github.com/TCNCoalition/TCN.
37. Trieu, N. et al. Epione: Lightweight contact tracing with strong privacy. *IEEE Data Engineering Bulletin 43*, 2 (2020), 95–107. http://sites.computer.org/debull/A20june/p95.pdf.
38. Troncoso, C., et al. Decentralized privacy-preserving proximity tracing. *CoRR* abs/2005.12273 (2020), https://arxiv.org/abs/2005.12273.
39. White, N. Creepy bartender uses coronavirus contact tracing data to ask out a girl he gave a free drink to—as Australians are warned their personal information could be misused or stolen. *Daily Mail Australia* (July 2020), https://www.dailymail.co.uk/news/article-8516533/Creepy-bartender-uses-coronavirus-contact-tracing-data-ask-girl.html.
40. Wymant, C. et al. The epidemiological impact of the NHS COVID-19 App. *Nature 594*, 7863 (2021), 408–412.
41. Zhao, Q., Wen, H., Lin, Z., Xuan, D., and Shroff, N.B. On the accuracy of measured proximity of Bluetooth-based contact tracing apps. In *Security and Privacy in Communication Networks 335*, Springer (December 2020), 49–60.

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/proximity-tracing