A Survey of Automatic Contact Tracing Approaches

LEONIE REICHERT and SAMUEL BRACK, Humboldt University of Berlin, Germany
BJÖRN SCHEUERMANN, Humboldt University of Berlin, Germany and Alexander von Humboldt Institute for Internet and Society, Germany

To combat the ongoing Covid-19 pandemic, many new ways have been proposed on how to automate the process of finding infected people, also called contact tracing. A special focus was put on preserving the privacy of users. In this survey we define classes of automated contact tracing techniques. We identify two major groups: systems that rely on a server for finding new infections and systems that distribute this process. Existing approaches are systematically classified regarding security and privacy criteria.

CCS Concepts: • **Security and privacy** → **Privacy-preserving protocols**; Mobile and wireless security; • **Applied computing** → **Health informatics**;

Additional Key Words and Phrases: Covid-19, contact tracing, privacy, survey

ACM Reference Format:
Leonie Reichert, Samuel Brack, and Björn Scheuermann. 2020. A Survey of Automatic Contact Tracing Approaches. 1, 1 (June 2020), 20 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Since the beginning of the year 2020, Covid-19 has turned into a global pandemic challenging both health care systems as well as democratic institutions [14, 23, 42, 78]. To mitigate its spreading, social and economic life was shut down in affected areas [77]. Tools often used in the past for containing diseases have proven to be not effective enough to deal with this quickly spreading, highly infectious and deadly virus [31, 73]. Therefore, new methods are developed to mitigate the pandemic such as to automate manual contact tracing done by health authorities to speed up the process of discovering new infections. Early systems implemented by Singapore, South Korea or Israel either used more data than necessary to fulfill the task or revealed to much information to the public [42, 74, 78]. In many countries, nationwide adoption of automatic contact tracing systems (ACT) applications cannot be enforced by the state [3, 24, 28, 46]. To ensure great effectiveness it is therefore essential that citizens trust the system enough to participate voluntarily. System designs that send detailed location or contact histories to a government-run central entity without any privacy protection might look more effective in the beginning. But societies will require transparent processes and data protection in exchange for their participation in the system.

Authors' addresses: Leonie Reichert, leonie.reichert@informatik.hu-berlin.de; Samuel Brack, samuel.brack@informatik.hu-berlin.de, Humboldt University of Berlin, Unter den Linden 6, Berlin, 10099, Germany; Björn Scheuermann, scheuermann@informatik.hu-berlin.de, Humboldt University of Berlin, Unter den Linden 6, Berlin, 10099, Germany, Alexander von Humboldt Institute for Internet and Society, Französische Straße 9, Berlin, 10117, Germany.

Many privacy-preserving ACT systems have been proposed, but the threats to mitigate are manifold. To compare the different currently discussed approaches we first provide background knowledge and introduce privacy definitions to assess and classify the different models.

The goal of this survey it to provide a general overview of different types of approaches for ACT by identifying two larger groups and several subgroups. We discuss shortcomings of each subgroup and problems common to all contact tracing systems based on Bluetooth Low Energy. In the next section, contact tracing and attacker models are introduced, as well as definitions that are used throughout the paper. In Section 3, ACT systems are discussed where an essential part of of the process, the risk evaluation, is run by a central server. Section 4 turns towards approaches where risk assessment is done on clients thereby decentralizing trust and computation. Servers are mostly used for relaying in these approaches. The section 5 deals with common security issues and threats of ACT systems, before we conclude with a summary in section 6.

2 CONTACT TRACING

2.1 Traditional Contact Tracing

Finding new cases by figuring out who had been in contact with a diagnosed patient has been used in the past for various diseases like HIV, SARS or Ebola [27, 84]. Both in theory and in practice it has proven to be a useful tool for containing epidemics. Stochastic modeling was used in [27, 44, 84] to evaluate the efficiency of contact tracing. An important result was that the rate at which new infections are discovered cannot be considerably lower than the rate at which the infection spreads [27]. A direct requirement for contact tracing following this finding is that possible contacts are notified as fast as possible so they do not infect others. Manual contact tracing is especially hard for airborne diseases like SARS, MERS or Covid-19 [27]. This is due to the fact that random encounters are difficult to notify: the diagnosed person can then oftentimes not provide all the relevant contact information.

2.2 Automated Contact Tracing

To ensure faster notification and to be able to notify random encounters it has become desirable to improve existing manual systems with modern technology in order to stop the Covid-19 pandemic [31]. In many countries smartphone apps are discussed for this purpose. These shall inform users of past risky encounters with people that were later diagnosed to ensure fast testing and quarantine.

Early research into the direction of automated disease transmission tracking was done by the FluPhone project [32]. The goal was to better understand and predict the influenza epidemic and how people alter their social behaviour in response. As part of the project a field trial was conducted [85], in which participants downloaded an app onto their phone that checked for other devices in the proximity using Bluetooth. For detecting phones close by, the FluPhone project built upon Haggle [71], a design for ad-hoc networks using Bluetooth. Information about encounters of devices was sent to a central server using mobile data. GPS measurements were used to improve results. Participants were asked to report symptoms using the app to determine if these indicated an influenza. The system also had the capability of marking devices as infected which could subsequently contaminate other users' devices they encountered based on probability calculations.

The first country to roll out a full ACT application for Covid-19 was Singapore with TraceTogether [37, 38]. Here, the app serves as support tool for the local health authority by speeding up their workflow. Users continuously send out pseudonyms using Bluetooth Low Energy (BLE) as shown in Figure 1. These beacons can be received and recorded

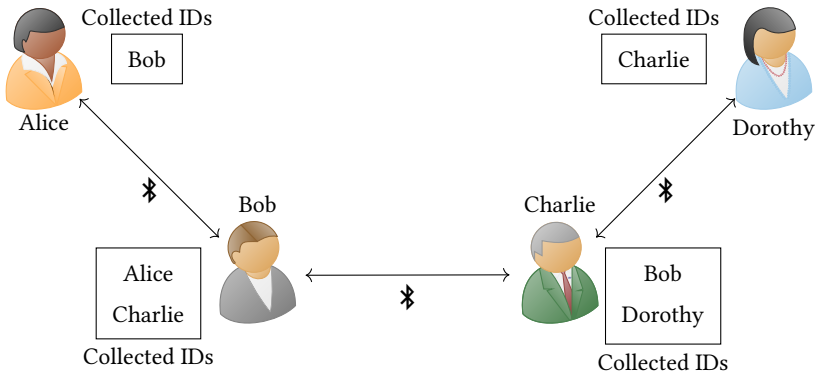

Fig. 1. During contact collection, each user stores the IDs of all devices that are in proximity. These IDs can be used to notify close contacts in case of a subsequently detected infection.

by other users. If a person is diagnosed, the beacons they have seen in the past are used to identify their random encounters. Most ACT systems also follow this BLE approach. The main difference between various approaches to ACT lies in the way how risk assessment is conducted and which parties hold relevant data.

2.3 Sensors

During the last years, Bluetooth has emerged as a useful technology for measuring the proximity between devices. First approaches to positioning and proximity detection using Bluetooth (especially indoors) were presented by [41, 50, 59, 70, 86]. These works use the receiver signal strength indicator (RSSI) to measure distance between receiver and transmitter and thereby derive a location. Raghavan et al. [66] were able to show that Bluetooth version 2.0 can be used for localization with an error of less than 45cm. Liu et al. [54, 55] demonstrated that Bluetooth is efficient for detecting face-to-face interactions by giving an model for estimating distance using RSSI readings.

Bluetooth specification 4.0 introduced Bluetooth Low Energy (BLE), an energy-efficient, short-range variant of standard Bluetooth [26]. In 2020, both Bluetooth and BLE have a high adoption rate, as 100% of new smartphones in 2020 support both standards [11]. Due to its battery saving properties BLE was adapted for positioning and proximity detection [29, 30, 57, 68]. Bertuletti et al. [9] were able to reduce the error of BLE measurements to less than 40cm. A major issue with using BLE for proximity detection and contact tracing is the large range in transmission power of different types of smartphones. RSSI readings therefore have to be calibrated to the respective devices [37].

Since using active sensors can open attack vectors, passive sensors like GPS [56, 67], Wifi [4], or Magnetometer readings [47, 60] can be used instead. Cell tower triangulation [78] is also an option for determining a person's location. In addition to Bluetooth, the Fluphone project tested RFID tags [32]. GPS data is generally seen as very sensitive, as it can reveal identifying information about a person like their home and work address. At the same time, its resolution is not fine grained enough to detect face-to-face interactions between people, especially in areas with tall buildings [48]. GPS also does not work reliably inside buildings. COVID-19 is an airborne disease, so while being in the same room as an infected person without protection is dangerous, sitting on the other side of a wall is not. These kind of false positive errors are difficult to mitigate when using GPS or cell tower triangulation. Wifi and Bluetooth/BLE are blocked by objects although the type of material plays a role [52]. While Wifi has been widely used for indoor positioning [52], just like cell tower triangulation it requires an infrastructure that might not be available everywhere especially outdoors or

in remote locations. Correlating magnetometer reading of users is passive and requires little energy while working indoors and outdoors. However, it only detects co-location at the moment and not the proximity between people. Two people might have recorded similar magnetometer reading at the same time. This means they were most likely at the same location but that does not provide any information about their distance to one another. There has been little research in this area so far.

2.4 Definitions

To ensure common understanding, we introduce the following definitions.

(1) *Automated Contact Tracing* (ACT) System: An ACT system consists of an app that can be installed on the users' mobile devices and a backend, typically a server. To function properly it is generally assumed that the local health authority operates the system.
(2) *User*: Users of an ACT system are people who downloaded the app and have it activated.
(3) *Infected* people: People are considered infected if their infection has been medically verified and reported. ACT systems can only consider infected people who have been using the respective system before they fell ill.
(4) *Encounter*: When two users Alice and Bob are in proximity of one another, this is called an encounter.
(5) *Contact*: If Alice is diagnosed as infected after an encounter with Bob, then Bob is called a contact of Alice.
(6) *At Risk*: Users are considered at risk if they have had encounters with infected people. This does not necessarily mean that they are infected.
(7) *Risk Scores*: Risk scores are calculated depending of the exposure of a user at risk. If the score exceeds a certain threshold, the user is notified.
(8) *Pseudonym*: BLE-based approaches advertise ephemeral or static IDs. Such IDs are called a pseudonym in this work.

2.5 Attacker Models and Types

When evaluating the security of a system, it is important to define the type of adversaries against which the system is secured. Attackers are generally distinguished into two types. Semi-honest, sometimes also called honest-but-curious, attackers follow the protocol but will try to learn as much information as possible. A malicious attacker has the additional capability to forge or replay traffic. The attacker can be computationally bounded or unbounded. It is also important to differentiate if an attack can only be conducted actively by communicating with the system or passively, and therefore with minimal interaction. Active attacks, such as trying out all possible inputs, are more resource intensive and easier to detect.

In ACT systems there exist several parties with different prior knowledge and capabilities:

(1) Health authority (HA): The public institution tasked with containing the spread of the disease. It may have an interest in learning as much about users and infected people as possible, for instance their relations to each other or where they have been in the past. Since infections with SARS-CoV-2, the virus causing Covid-19, have to be reported in many countries [13, 39], it can be assumed that the HA possesses a considerable amount of information about infected users. In some legislations it is even a crime to not support the HA during contact tracing [39]. The HA does not have an interest in blocking contact tracing or stopping risk notifications to users.
(2) Users: Users want to determine their health status. They might also have an interest in figuring out which of their social contacts is infected or who infected them. A type of user called *Curious Stalker* or *Paparazzi* stalks a

target in order to find out if this person is infected. The stalker can follow her victim and observe if his habits change.

(3) Infected people: Infected people participate in most systems through having been reported to the HA by their doctor. They have an interest to not reveal too much sensitive information about themselves to the public and the HA, because they fear public humiliation [72] or other forms of social outcasting.

(4) Service operator: The ACT service and its infrastructure can be run by the HA or by a third party such as a contractor. Servers and cloud storage fall into this category. A service operator can try to learn general information about users and infected people as well as their health status by observing and manipulating data passing their system.

(5) Network operators: Network operators can have similar goals as service operators, but are only capable of observing and manipulating data that is sent through the network.

3 SERVER-SIDE RISK ASSESSMENT

Numerous ways exist how the infection risk of users in ACT systems can be calculated. From a structural perspective, risk scores can be either determined on the server or on the client. Both approaches come with different security risks and trust models. In this section, ACT systems using a central server for risk assessment are discussed. Since infectious diseases are subject to mandatory reports to the HA in many countries, it is a natural candidate to run central infrastructure for ACT. The systems discussed in this section mostly rely on the HA to collect data from infected individuals. The HA ensures that all collected data is legitimate. This is an important step, as false claims of infection could cause fear and chaos within affected communities.

3.1 Results Revealed to Server

The first widely deployed ACT system has been developed for the government of Singapore [37, 38]. The app is called TraceTogether, while the associated open source project has the name Bluetrace. End devices of users run an application which uses ephemeral BLE beacons to advertise their presence. These pseudonymous beacons are generated on the central server, so that the server knows at all times which pseudonyms belong to which user. After some time a new pseudonym is broadcast to ensure that users cannot be tracked by a third party other than the HA. The app

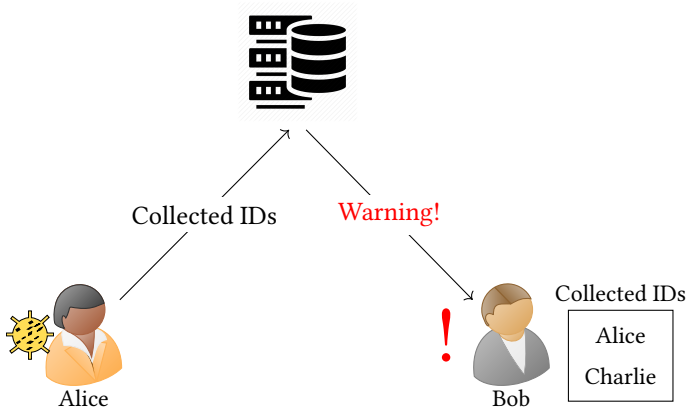

Fig. 2. Here the general idea of server based ACT is visualized. Alice sends her collected IDs to the server when diagnosed as infected. The server does a risk assessment for her contacts and warns Bob.

also continuously scans for nearby devices that advertise themselves. If another device is registered, the announced pseudonym of the other user is stored locally for a predefined period of time. Depending on the disease, the retention period can be different and is derived from epidemiological findings. In case of Covid-19, pseudonyms are stored for two to three weeks. As soon as a user Alice is diagnosed with the disease, she uploads her history of observed pseudonyms to a central server. The central server performs a lookup for all collected pseudonyms to re-identify users and calculates their individual risk scores. Risk scores can be influenced by factors like the duration of the encounter, the signal strength of the transmission indicating proximity, or the number of infected users that reported a contact with the user at risk.

After determining the individual risk score, an assessment is performed by the server (see Figure 2). If a certain risk threshold is exceeded, the server will notify users that are at risk. Following this notification, affected users are requested to place themselves under medical care or into immediate quarantine.

A very similar concept to TraceTogether can be found in the framework of PePP-PT [62]. PePP-PT is a European initiative that focuses on a centralized approach as well. Similar to TraceTogether, the central server is operated by a country's health authority. Pseudonyms for BLE are generated by the server and sent to user's device which announces them over its Bluetooth interface. These pseudonyms are encrypted values of the users' fixed ID. If an infected user Alice reports herself to her country's HA, she can transmit her list of collected pseudonyms from the last 14 days to the corresponding server. Each of Alice's collected pseudonyms can be decrypted by the server that issued it and the individual risk scores of the users at risk can be calculated. Users at risk are then notified with push notifications.

Two implementations of PePP-PT exist, PePP-PT NTK [63] and ROBERT [45]. They only differ in minor details. For instance, ROBERT uses 3DES as their symmetric encryption algorithm instead of AES. To facilitate cooperation between different states, both TraceTogether and PePP-PT allow for cooperation between different health authorities.

The described models are very similar in their operation and have the same advantages and disadvantages. Using this central approach, the identities of people who should quarantine are revealed to the HA and restrictions can be enforced. Also no data is revealed to users other than the risk notification received by affected users. When a risk notification is received, a user can only guess that they might have been infected by someone from their history of encounters. But since proximity measurements are made independently both parties might record different distances and an encounter might have only been recorded by one side. So simply using the own history of encounters when trying to figure out who is the cause for a risk notification is not reliable for an attacker. This means this type of approach protects the identity of infected individuals well against other users. Instead, the dangers of a centralized ACT system lie elsewhere as information about the relations of users leaks. In case a user is reported as a contact by several infected patients, the server can directly derive that these people might know each other. It also learns about relations between uninfected users as the HA can observe that some users always appear at the same time in collected data sets. Using additional information such as the time of an encounter or other prior knowledge, the HA can find out specific details about the nature of users' relations. While these individual relationships might seem insignificant, this attack vector allows the adversary to build a social graph for parts of the user base.

A malicious HA could even install Bluetooth sensors in popular areas like train stations and collect pseudonyms there. This allows the HA to learn the location history of any user who passes the capture device, as it knows who is using which pseudonym at what time. Depending on how tightly knit the infrastructure of publicly located Bluetooth sensors is, the HA can follow every movement of users.

Another issue arises from the way how ephemeral pseudonyms are linked to static ones at the backend. For example in PePP-PT, ephemeral pseudonyms are created by encrypting a static identifier. The reference implementation of

Bluetrace works similarly. If the encryption key is leaked, all identifiers issued with this key become linkable and recorded traces can be deanoymized. It has been proposed to use rotating keys to reduce this threat [83].

As explained in the introduction, it is essential that users trust the contact tracing system enough to participate voluntarily. Many people seem to be deterred by systems they find too intrusive or incapacitating [5], such as one where they are forced into quarantine instead of taking the decision themselves. There is also the fear that centralized approaches facilitate the creation of new surveillance infrastructure that could, for example, be used to target minorities [5, 14, 51]. These two aspects have greatly influenced the public discussion in some European countries causing governments to move away from centralized approaches as described above [19].

3.2 Using Cryptographic Building Blocks

Some approaches to ACT allow risk assessment done on the server or in collaboration with the server while revealing the risk score only to the affected users themselves. These approaches leverage modern cryptographic tools such as homomorphic encryption and secure multiparty computation (MPC).

3.2.1 Homomorphic Encryption. Homomorphic encryption (HE) [58] describes encryption schemes which allow computation on already encrypted data. A homomorphic function is defined as follows: Let $f(x_1, x_2, \ldots, x_n)$ be a function with n inputs. A function h is a homomorphic encryption function of f if for an encryption function $e(x)$ and the corresponding decryption function $d(x)$ it holds that $d(h(e(x_1), e(x_2), \ldots, e(x_n))) = f(x_1, x_2, \ldots, x_n)$. In fully homomorphic schemes, encrypted data can be added or multiplied as often as necessary to construct arithmetic circuits. The decrypted result will be meaningful and reflect the result of the operations conducted on the encrypted data. Some homomorphic encryption schemes are limited in the amount of operations on an encrypted set of data and utilize a noise budget, where each operation draws from until the operations become unreliable when the budget runs empty.

An early approach to privacy-preserving ACT is the EPIC framework [4]. Here, users do not actively send out pseudonyms but passively fingerprint their surrounding by capturing both Bluetooth and WiFi beacons. Such location fingerprints captured by infected users are uploaded to servers belonging to the HA. Uninfected users send requests to the server to determine how similar their location fingerprints are to those measured by infected users for certain timestamps. The request contains the public key of the user, an encryption of the location fingerprint at timestamp t_e, and t_e. The server will use the provided public key to encrypt location fingerprints with a close timestamp and then calculate a matching score. The scores cannot be decrypted by the server. It will send the result back to the uninfected user who can decrypt it and derive their personal risk score.

Another approach using homomorphic encryption was proposed by Bell et al. [7] as part of TraceSecure. This system relies on a BLE-based exchange of pseudonyms. This scheme reveals to the server who has interacted with whom but aims to keep the health status private.

3.2.2 Secure Multiparty Computation. The field of *secure multi-party computation* (MPC) [75, Chapter 22] deals with creating protocols for joint computation on private, distributed data. It studies mechanisms to allow a group of n independent participants to collectively evaluate a function $y_1, \ldots, y_n = f(x_1, \ldots, x_n)$. Each participant holds a secret input, which remains hidden to other parties but is used for computation. The participants only learn their designated final result. Any function f that is solvable in polynomial time can be represented as an MPC protocol [75, Chapter 22.2]. For ACT generally only two parties are considered, a server and a client trying to determine its risk status. One way of realizing arbitrary MPC protocol are *Yao's garbled circuits* [75]. Running an MPC protocol using this technique requires one side to create a *circuit* from the function to be calculated and send it to the other party. The other side evaluates

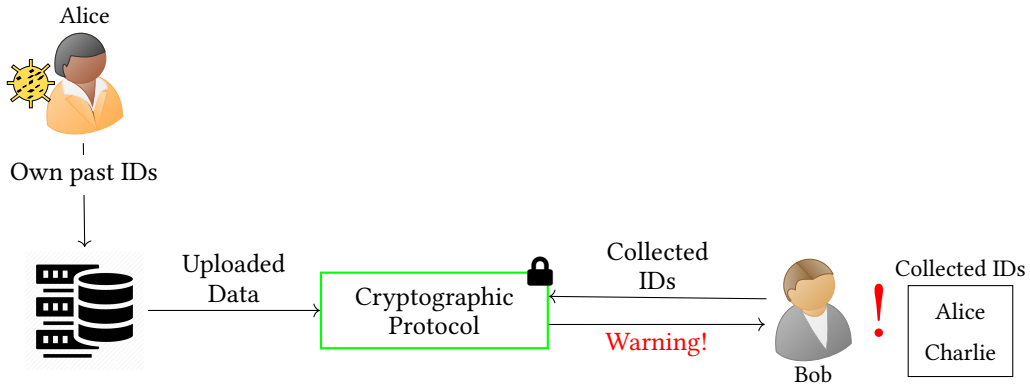

Fig. 3. This figure illustrates how a simple private set intersection protocol for BLE-based ACT could work. This example does not leak the intersection to Bob.

the circuit. Evaluation requires oblivious communication between both parties. The smallest MPC building block are *oblivious transfers* (OT), where one side offers two values and the other can select on of these using an index without learning the input of their counterpart.

One application in MPC is private set intersection (PSI). The two participants each hold a set of elements and want to calculate the intersection without revealing elements not contained in the intersection. This type of protocol can easily be mapped to the problem of privacy-preserving ACT (see Figure 3) .

Berke et al. [8] used Diffie-Hellman private set intersection. Instead of exchanging Bluetooth IDs, the authors use GPS traces. Coordinates are truncated and rounded so that they are represented by single dots on a three-dimensional grid (longitude, latitude, and time). Since distance is an important factor when transmitting the virus, for each truncated coordinate it is also important to check whether the neighbouring grid points are part of the intersection. To execute Diffie-Hellman PSI on the set of grid points, both client and server first need to create an asymmetric key pair. Each side encrypts their set with their private key and sends it to the other. They then encrypt the already encrypted set with their key, so now each set is encrypted with both private keys. The server sends the set it encrypted last also to the client, which then holds both sets. The client calculates the intersection of these encrypted sets. Due to the multiplicative property of asymmetric encryption, it is not important which key was used first. This protocol can be used to allow clients to learn the size of the intersection, but also which of their elements appear on the servers by letting the client query for elements individually. An approach by Reichert et al., also for GPS data, works similarly. Instead of using PSI, binary search on oblivious memory is used to determine if an element appears in the server's set of infected users' location data [67].

The protocol of Dimrag et al. [22] uses Bluetooth to advertise a static ID. The HA server holds all IDs of people with verified infections. To figure out how many people they have met in the last weeks that were infected, the user performs private set intersection following the protocol of De Cristofaro et al. [20].

Epione proposed by Trieu et al. [79] also uses BLE to exchange ephemeral pseudonyms. They use Diffie-Hellman based PSI algorithm to determine the cardinality of the intersection. The algorithm is optimized for situations where the client's set is a lot smaller than the server's set. This approach also uses homomorphic encryption for some steps.

The approaches discussed in this section are cryptographically secure, meaning they leak no more information than intended by the protocol. All MPC protocols can be secured against malicious attacks by accepting performance

penalties [34]. Runtime and communication remain problematic in these designs. Circuits can become very large and may require many gigabytes of data to be communicated. This is hardly feasible on mobile data connections. Research on PSI does exist that attempts to take load off the end devices [49]. Still, DDoS against the central server remains a problem. Also, attack vectors based on data of infected individuals – such as their estimates based on their location history or leakage of the social graph based on published pseudonyms/IDs – remain as challenges that need a solution before such an approach is feasible in a real-world setup.

4 CLIENT-SIDE RISK ASSESSMENT

A different type of approach is based on the idea that the risk status of a user should be calculated locally on the client's device and not be revealed to the HA, service providers, or network providers. This often requires more resources on end devices. In some literature this category of ACT is also called *decentralized*. Several such models with client-side risk assessment for ACT are discussed in this section, where we distinguish between systems using broadcasts and ones using direct transfer of messages. A simplified illustration of the idea is displayed in Figure 4.

4.1 Broadcast Models

DP-3T[80] is one large initiative using the broadcast approach. In DP-3T's early so-called low-cost approach users use an individual seed to derive a daily key. This daily key is then used to deterministically calculate rotating BLE pseudonyms called ephemeral IDs. When a user becomes infected, the daily keys for the relevant time period are uploaded to a central server and distributed to all users. The user application then locally derives the corresponding ephemeral IDs of the infected user and checks in its history if there has been an encounter with any of these. A major problem with this approach is the fact that an infected user's identity becomes linkable over the two weeks before the infection. To mitigate such attacks by curious users or eavesdroppers, DP-3T developed a second approach called the unlinkable design. Here, for each time slot a new completely independent pseudonym is generated. When a person becomes infected, the pseudonyms are uploaded to a server which stores them in a global Cuckoo hash table [81]. Users

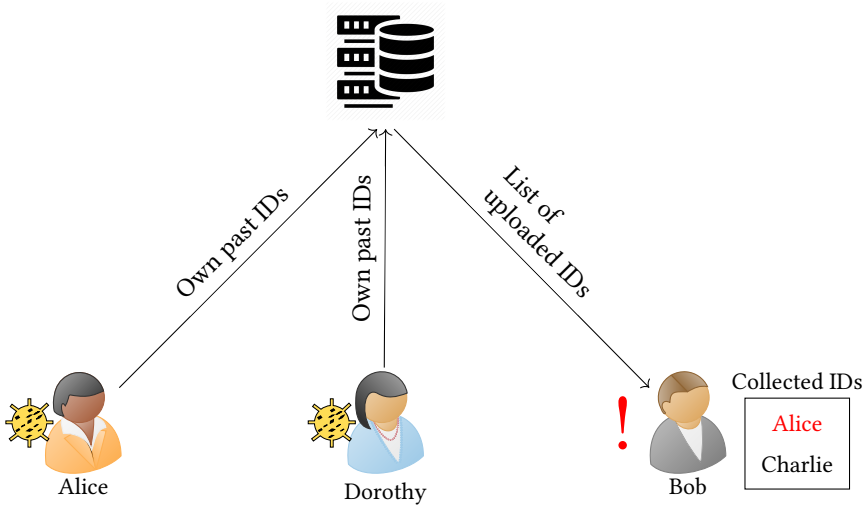

Fig. 4. This figure visualizes the idea of a simple broadcast ACT model. When Alice is verified as infected, she will upload the IDs she advertised in the past to a server. Bob will download a list from the server containing Alice's IDs, but also those of other infected users such as Dorothy. Checking locally against the list, he recognizes one of Alice's IDs.

will download the hash table regularly and check if any of their past encounters causes a hash collision. To ensure that the failure probability of the hashing process remains low, the server creates a new, empty table after some time [16].

Apple and Google, as the companies jointly dominating of the market for smartphone operating systems, formed an alliance to present a joint approach for ACT [36]. They propose a technical specification for an API using the ideas of DP-3T. Differences between DP-3T and the Google/Apple tracing scheme are mostly on an implementation level. While DP-3T derives the daily key by hashing the key from the day before, Google/Apple combine the initial tracing key with the number of the day in a key derivation function. Another difference is how ephemeral IDs are created. DP-3T derives one value for a whole day by feeding the daily tracing key first into a pseudo-random function like HMAC-SHA256 and then using the result as the input for a stream cipher like AES. Then the output is split into chunks of 16 bytes and shuffled before usage. Google/Apple derive ephemeral BLE IDs independently by feeding the daily tracing key and the number of the current time interval into a pseudo-random function. The result is 16 bytes long and is used immediately. Concerning realisation of ACT, the two companies insist on only providing an application for end devices but leave setting up server infrastructure to HAs interested in cooperating. Other examples for a similar scheme are CONTAIN [43], PACT by Rivest et al. [69] and PACT by Chen et al. [15]. The latter does not publish pseudonyms of infected users but instead secret ephemeral IDs of contacts. For each time slot a user has a public and a secret ID, which are both communicated to the server. The server will later lookup secret IDs by using the history of encounters of infected individuals.

Covid-Watch [18] is a project supported by the University of Stanford also working on a broadcast approach. Instead of ephemeral pseudonyms, a new random number is generated per contact event. Another difference to the two projects mentioned above is that when a user is tested positively, they will not only upload their own number used in the past but also the BLE pseudonym of others. This information is then broadcast to all other users who then check locally if they have a corresponding contact event stored locally.

Pinkas and Ronen proposed an elaborate key derivation mechanism for ACT systems using a broadcast mechanism [65].

Approaches using the broadcast model are able to hide the fact that someone has been in contact with a person who tested positively. This can be an important feature to gain users' trust, as they are able to review warnings for plausibility and are free to decide for themselves when it is time to get medical attention. Since the risk status is calculated locally and all users receive the same data, service providers and network providers cannot guess a persons health status by eavesdropping. Broadcast models have the common weakness of revealing the pseudonym and approximate time when the contact occurred. Overly curious users could try to abuse this information to deanoymize infected people. This also simplifies attacks where a security camera is combined with a Bluetooth sensor device. Here, the captured data allows the attacker to connected infected pseudonyms to faces. Another issue are impersonation attacks. An infected user could upload different pseudonyms than the ones they used themselves to make it seem like someone else is actually infected.

4.2 Direct Messaging

Another way of doing risk assessment for ACT on client devices are postbox systems instead of using broadcasts. The approach was first described by Cho et al.[17] (see Figure 5).

Here, users regularly create a new asymmetric key pair and use the public key as ephemeral BLE pseudonym. The private key is stored locally. When a person has contracted the disease, they use the collected IDs of other users to notify them. To do so, they place a message encrypted with the other user's pseudonym into the corresponding postbox.

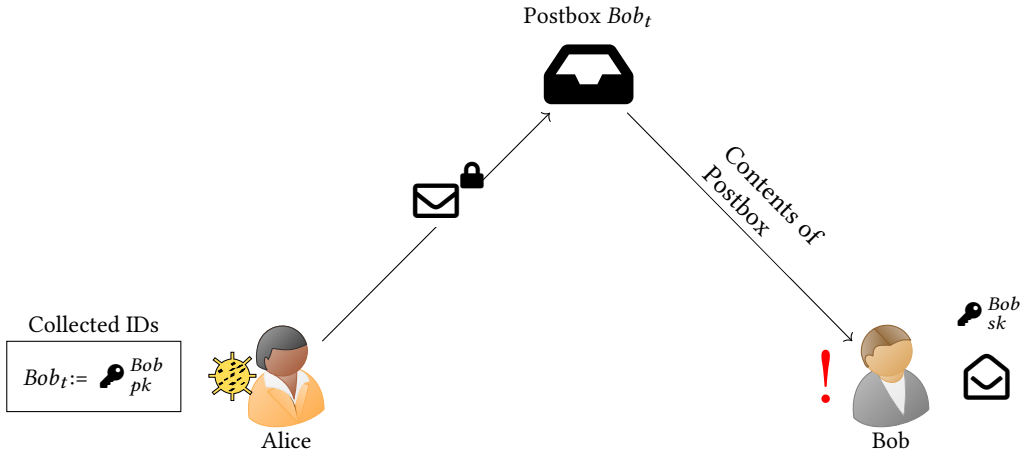

Fig. 5. An example of a direct messaging approach to ACT following Cho et al. [17]. Alice collected Bob's ID Bob_t which she uses to encrypt a message for Bob. This message is placed in the corresponding Postbox where Bob can retrieve it. After its decryption he knows that he is at risk.

Users regularly check postboxes belonging to their past advertised keys to see if a new message has arrived. To ensure that the server cannot link real identities with postboxes, the authors require requests to the server to be sent through a network of proxies. To mitigate deanonymization by observing traffic it is also necessary to introduce cover traffic. This means users not only send messages to others when they become infected but also send messages stating that they are still healthy. One issue not discussed in this proposal is the aspect of authenticity. Users can try to cause panic by sending "I am infected" messages to many people without actually being at risk. A system proposed by Brack et al. [12] is based upon the approach described above and solves this authenticity problem. Here, blind signatures are used to ensure that only sick users are able to warn others. When a user is infected they blind their own past IDs for which they need a signature and individually send them to the HA. The HA signs the blinded message it receives and sends is back. It does not learn the plaintext message value during this process. The user can then unblind the returned values and now holds valid signatures for their own IDs. This step has to be additionally secured using, for example, permission keys which an infected user requires to authenticate their health status to the HA. Each permission key can only be used for one blinded message to prevent linkability of signatures. Permission keys could be issued by the local doctor when test results are positive. At-risk contacts are notified by the infected user by placing an encrypted message in the postbox corresponding to the contact's advertised ID. The message is encrypted to the contact's ID (which is also a public key) and contains the signed ID the infected user advertised at the time of the encounter. The designated receiver collects the encrypted message, decrypts it locally, and validates the signature inside by using the HA public key. For giving users access to their postboxes, a distributed hash table is used. While this allows to completely remove the central server, it creates new attack vectors that did not exist before, such as Sybil attacks or adversaries trying to gain control over specific mailboxes.

A system similar in concept to direct-messaging ACT systems is Pronto-C2 [6]. Users derive a shared key from the advertised pseudonyms that is only identifiable to someone in possession of both pseudonyms. If someone is infected, the generated key is published. Users regularly search for keys on the server. The authors did not consider the need for cover traffic, so while users do their risk scoring locally, this system leaks information. The server learns possible keys

and eavesdroppers can figure out from the response whether a querying person is infected. The authors propose to use a blockchain for the server to ensure that no already published data can be deleted.

The message-based ACT protocol of TraceSecure [7] take up these ideas of Cho et al. Their protocol relies on multiple non-colluding parties: the HA, the government and for some cases a messaging service. When joining the system, users have to (anonymously) send their seed used to derive ephemeral BLE IDs to the government. In return the user is provided a static pseudonym which they can use to check with the messaging service if new messages have arrived. When a user is diagnosed as infected, they notify all past contacts individually by having the HA relay encrypted messages to the government. Since the government knows the seeds it can derive which static pseudonyms need to be warned. It places corresponding messages in the messaging service so users can receive them. This system requires cover traffic on the path from the government to the user. Since the HA holds the seed for all users, she can derive a user's current advertised BLE ID and use this information for tracking.

Another example for a direct messaging approach is ConTra Corona [10]. Pseudonyms advertised by users are the public part of an asymmetric key pair. The private key is uploaded to a so-called matching server. If a user is tested for Covid-19 and their test comes back positive, medical personnel forwards the recorded pseudonyms in encrypted form to the matching server. The server looks up the corresponding secret key and marks it as infected. Users can either query for their private keys or the matching server publishes them regularly. This approach relies on multiple non-colluding central parties.

Deducing health status from traffic patterns is a big problem for all postbox systems, therefore cover traffic is required. But allowing arbitrary traffic makes mitigating spam difficult. Attackers can try to congest a specific postbox so that the corresponding user will not be able to receive valid messages for this ID.

5 COMMON SECURITY ASPECTS

In this section we discuss common security and privacy aspects that can potentially arise in all ACT systems. These aspects range from hardware-specific threats to privacy leaks triggered by targeted attacks.

5.1 Bluetooth

Since Bluetooth/BLE is the base of most ACT systems, we will discuss problems and attacks against this approach in more detail.

5.1.1 Jamming. Companies or individuals wanting to stop contact tracing on their premises can block the exchange of pseudonyms by jamming the respective channels.

5.1.2 Storage and Power Drainage Attacks. Another simple kind of attack targets the exhaustion of battery power and storage of the end device by sending large amounts of BLE beacons [40]. This might make the ACT system unappealing to users, hindering widespread adaption.

5.1.3 Linking Advertisements. By measuring the time between the announcements of pseudonyms, it is possible for an attacker to find out which successive pseudonyms belong to the same person. It has been proposed to add jitter to the intervals between announcements [10, 40]. When an end device advertises itself, a MAC address is also part of the transmission. This MAC address changes regularly. To ensure that linking of different pseudonyms of the same person is not feasible, it is important to change the MAC address at exactly the same time when the pseudonym is rolled over to its successor. This feature requires support by the operating system and has been announced by Google

and Apple [35]. Another point when trying to mitigate linking attacks is to consider the RSSI data. These proximity measurements allow an attacker to determine if two successive pseudonyms originated from the same approximate location. Gvili [40] proposes to have senders vary the signal strength in a way that makes it difficult to deduce the location of a user from only few samples.

5.1.4 Passive Eavesdroppers. A passive eavesdropper can collect pseudonyms over BLE and use these later to deanoymize users. An attacker needs enough financial resources to install the required infrastructure in highly frequented public places. Some infrastructure might already exist due to digital billboards being equipped with BLE sensors. This attack works especially well in decentralized systems such as DP-3T, where used pseudonyms of infected individuals are simply published. Users can also be deanonyminized by linking their successive pseudonyms and deducing their daily movements.

One mitigation measurement is beacon secret sharing as proposed by the authors of DP-3T. Here, instead of advertising the pseudonyms, only fragmented shares of pseudonyms are broadcast. The other side need to collect a certain number of shares to deduce the sender's actual pseudonym. It therefore becomes difficult for a publicly located Bluetooth device to receive meaningful pseudonyms from people who are simply passing by. Another approach to stop a passive eavesdropper is to tie the risk notification to the requirement of both sides exchanging pseudonyms and registering the encounter. This is especially enforced by direct messaging approaches.

5.1.5 Active Eavesdroppers. An attacker might not be satisfied with passively collecting IDs and instead equip each of their BLE devices located in public spaces with the targeted contact tracing app. This way, a passing user will also collect an ID of the attacker's BLE device. Since public places are usually crowded and most systems change pseudonyms regularly, detection is unlikely. Even worse, if security cameras are equipped with ACT applications, exchanged pseudonyms can be linked to surveillance footage. This makes infected individuals easily deanoymizable at a later point in time using corresponding pictures. Again, secret sharing of beacons does help against attacks on users who are at the location only for a short period of time. The number of shares is an important parameter to consider, as more shares means higher privacy, but also might harm utility.

5.2 Impersonation Attacks

Apart from attacking the physical Bluetooth layer, an attacker can also try to gain sensitive information by impersonating others.

5.2.1 One Contact Attack. Assume an attacker wants to find out if a person will later be infected. They could create a new account just for an encounter with this user. If they later receive a risk notification, the attacker knows that it was this specific person who triggered it. One way to mitigate this attack would be to make the creation of a new account difficult for example by installing captchas or tying it to a phone number. Another solution discussed by Gvili [40] is to ensure that a user is always protected by k-anonymity. If less than k distinct BLE advertisements are detectable, end devices create cover traffic to make it look like more users are in the general area. A passive observer will not be able to determine which transmissions come from which users, especially if the signal strength is varied.

5.2.2 Replay and Relay Attacks. Another problem of the BLE approach is that of an attacker recording pseudonyms and replaying them at a different location. The attacker can for example collect data in a high-risk environment such as a hospital and play it out in another location like a cafe frequented by a certain target or demographic. To limit the impact of replay attacks most approaches [12, 80] in some way encode the epoch of the encounter in the transmitted

pseudonym. Centralized systems like TraceTogether can check when an encounter was recorded and whether the recorded pseudonym was actually in use at that time. Broadcast systems allow users to check themselves if they recorded a corresponding encounter for this time slot. Direct messaging approaches can encrypt the sender's pseudonym at the epoch of the encounter to allow the receiver to do a similar check. This requires (loosely) synchronized clocks, but even deviations of several minutes are acceptable. The situation is different when the attacker replays the collected pseudonyms during the same epoch in which they were collected. As mitigation against this kind of attack [82] proposes to switch from passively exchanging pseudonyms through broadcasts to an active protocol. It has been warned that an active exchange of messages is more insecure than one-way communication where users simply send advertisements and listen to other advertisements. Using an active exchange opens the door for new types of attacks against the end device. Energy consumption also increases in such a scenario. Some works [40, 64] propose to use coarse (GPS) location data in the broadcast of the pseudonyms. This allows the receiver to figure out it the sender is actually close.

5.3 Authentication and Verification

ACT systems require some kind of interface to the testing infrastructure and to the users to distribute meaningful risk notifications. Trolling and spam need to be prevented to ensure the system is useful.

5.3.1 Authenticating Uploads. Ensuring that a user is indeed infected while enforcing privacy is an important aspect. Simply not controlling who is capable of uploading allows for trolling and makes the system unreliable. Having infected individuals simply upload all their data as done in centralized systems leaks information about social interaction. Most ACT approaches use token systems that allow infected users to upload their data to the server after having received confirmation from a doctor. To prove to users that data was sent by infected individuals some direct messaging approaches [6, 12] use blind signatures. This way the health authority does not learn the content to be signed, but users can fetch valid signatures. This provides certainty for message receivers in these systems. In this situation, a hacked malicious server (without access to the HA's private key) can only delete messages but not insert new ones.

5.3.2 Verifying Encounters. Imagine a black market where people offer money for faking contacts of a target person with infected persons. Someone who knows they are infected can alter the data they upload so it looks like they have been in contact with the target. In BLE-based ACT systems this can be stopped by having the client check if they have recorded a corresponding contact event. Liu et al. [53] take a different approach. When users have an encounter of meaningful duration (e.g., 15 minutes) they initiate an active exchange over Bluetooth to swap identifiers and signatures. Later, zero-knowledge proofs are used to demonstrate to the HA that an encounter actually occurred.

5.3.3 Incomplete Reports. Users want to have control over what they report, so that no sensitive data is leaked. For this purpose some systems provide the option for users to opt out of uploading some or all data to the server. This leaves room for extortion, as infected people could blackmail other users for not being included in the infected person's upload.

5.4 Metadata

An important aspect of operational security is to check whether metadata can leak information that is intended to remain secret.

5.4.1 IP Address Leakage. Many ACT systems rely on the IP address not to be leaked when communicating with central infrastructure. Users of a system where risk assessment is done on the server might have an interest in not revealing

their identity directly to the server, although centralized systems like PePP-PT might be able to deduce it based on other metadata. In decentralized systems users might not want to reveal the fact that they participate. Depending on the actual authentication mechanisms, users might want to ensure that uploaded data (like past pseudonyms) are not linkable to their identity by the HA. The security assessment of decentralized systems generally relies on proxying to ensure that no single party learns the real identities of sender and receiver. For this purpose anonymisation networks like Tor [25] or mix networks [21] can be used. If users use such a network when communicating with the server, it will not learn their real IP addresses (and thereby their identity) as they are hidden by a cascade of proxies. While Tor-like anonymisation infrastructure is vulnerable against timing attacks conducted by adversaries capable of monitoring large parts of the network [61], mix networks are hardened against this type of attacker, but are slower at delivering messages.

5.4.2 Leakage through Timing. Other metadata that might be used to derive information is time. When uploading data that should not be linked by the server, it is necessary to also induce jitter.

5.5 False Positives and False Negatives

An issue often mentioned when discussing the applicability of contact tracing are false positives and false negatives.

5.5.1 False Positives. A false positive in the case of contact tracing can belong to one of two categories. One option is that the situation for an encounter that did not occur at all. In the other case the ACT system detected an encounter even though the transmission of the disease is highly unlikely, e. g., when two users were separated by a wall. Reasons for such errors can be manifold. To minimize the number of false positives based on distance, one option is to lower transmission power or improve the model for distance estimation, e. g., by having the sender provide information about its current transmission power or by calibrating the sender. To ensure that an encounter was actually relevant, it is important that only those with a significant time span are taken into account. Some contacts might have not been relevant as they occurred outside when it was windy so the infectious aerosol was dispersed. End devices can make use of all available sensors to heuristically determine if an encounter took place indoors or outdoors. When using GPS measurements, weather data can also be taken into account when doing risk estimations. For systems doing risk evaluation on the end device such extensions are easily applicable without endangering the users' privacy.

5.5.2 False Negatives. Risky encounters might not be detected causing infected users to not be warned by the system. Here, the solution would be to increase transmission power while ensuring that other measures are in place to mitigate false positives. The balance between both types of errors is important.

5.6 Proving Risk

It has been suggested to ensure that users who have received risk notifications have a right to be tested. This is especially of interest in places where testing capacities are sparse. In centralized systems such as TraceTogether it is easier to determine who is eligible for a test, as servers provide some degree of validation. Infected users altering their history are a risk to all systems known to the authors. For decentralized systems it is not as simple. Even if a user receives a notification, they have to prove they actually had a contact and are not simply forging encounters just to get tested. For direct messaging apps relying on asymmetric key cryptography, the possession of a private key corresponding to an at-risk public key can be used as proof. To prove exposure Hashomer [65], which falls in the group of broadcast approaches, derives one part of the advertised pseudonyms from a verification key. This key is later uploaded to the HA.

Users that want to prove they are at risk can present the corresponding collected ephemeral IDs. Using the verification key, the HA can establish if the collected ID belongs to an infected person. This approach opens up new ways for the HA to derive relations between users and does not prevent the transfer of known infected pseudonyms to other users. The authors of ConTra Corona [10] propose to incorporate a random value u into all pseudonyms that can later be presented in a non-interactive zero-knowledge proof of knowledge to verify ownership. To discourage people for giving away their proof, u can include a timestamp and the user's real identity.

The Corona-Warn App created by SAP and Deutsche Telekom for the German state aims at providing an interface between testing infrastructure and tracing infrastructure [2].

5.7 Hacking, Backdoors and Malware

Common ACT systems rely on apps installed on smartphones. Like in any kind of IT environment, both underlying hardware and software can be vulnerable.

Users' trust is an important building block of ACT systems. It has therefore often been mentioned that making code open source is a requirement. This allows independent security researchers to check that no back doors exist and that the app is not actually malware. Additionally, independent audits would be necessary to ensure that it is the same open source code running on the backend servers and in the application. To ensure that no other installed applications can spy on the ACT app, it has been argued that employing Trusted Platform Modules (TPM) would help [82].

5.8 Bluetooth Vulnerabilities

Since devices advertise themselves, they signal to possible attackers where to find a device with activated interface, who can then exploit known vulnerabilities such as [1, 33] to gain unauthorized access to users' devices. Pairing of devices needs to be avoided to mitigate the relating additional risks such as [33, 76]. The only working mitigation against this kind of attack is to regularly apply security patches or to use passive sensors for proximity detection such as GPS or the magnetometer.

5.9 Dealing with International Travel

To facilitate cooperation between different states, PePP-PT includes a system for federation between different health authorities [63]. A country code is added to the encrypted ID. TraceTogether also supports federation, in a similar manner [38]. Decentralized ACT systems can also support federation, if the app allows downloads from servers of other countries also using the decentralized approach. For systems using direct messaging, no options for federation have been discussed.

6 CONCLUSION

In this paper we classified automatic contact tracing systems based on where risk scoring occurs. Table 1 provides a compact overview of all discussed approaches. For centralized approaches we distinguished between approaches revealing the risk score to the server and systems that use cryptographic primitives such as MPC or HE to ensure the users' privacy. For ACT systems where risk scoring is done on the end devices we identified the broadcast model and the direct messaging approach. For all groups we identified common attack vectors and discussed mitigation measurements. It remains to be seen if automated contact tracing lives up to the expectations and how feasible the different types of systems are in real-world settings.

Table 1. Overview of contact tracing approaches. "(1)": Authors did not differentiate between BLE and Bluetooth. "(2)": For known pseudonyms. "(3)": Cryptographic overhead on end devices. "(4)": Cryptographic and polling overhead on end devices.

		Name	Base technology	Trust model for server	HA can track users	Results revealed to HA	Infected users deanonymizable	Computation intensive	Traffic flow analysis for to find infected users	Notes
Server-side	Central	TraceTogether/ BlueTrace [38] PePP-PT (NTK [63], Robert [45])	BLE	Trusted	x	x			x	
			BLE	Trusted	x	x			x	
	Cryptographic	EPIC [4]	Passively collected Wifi+Bluetooth advertisements	Semi-honest				x		HE
		TraceSecure (HE approach) [7]	BLE	Semi-honest				x		HE
		Berke et al. [8]	GPS	Semi-honest			(2)	x		MPC, PSI
		Reichert et al. [67]	GPS	Semi-honest/ malicious			(2)	x		MPC
		Demirag et al. [22]	Bluetooth(1)	Semi-honest				x		MPC, PSI-CA
		Epione [79]	Bluetooth/BLE	Semi-honest/ malicious				x		HE+MPC, PSI-CA
Client-side	Broadcast	DP-3T [80]	BLE	Semi-honest			(2)	(3)		
		Apple+Google [36]	BLE	Semi-honest			(2)	(3)		
		CONTAIN [43]	BLE	Trusted			(2)	Proto 1: (3)		
		PACT (Rivest et al.) [69]	BLE	Semi-honest			(2)	(3)		
		PACT (Chan et al.) [15]	Bluetooth(1)	Semi-honest			(2)	(3)		
		Covid-Watch [18]	Bluetooth(1)	Semi-honest			(2)	(3)		
		Hashomer [65]	BLE	Semi-honest			(2)	(3)		
	Direct messaging	Cho et al. [17]	BLE	Semi-honest			Gives time period of encounter	(4)	Uses cover traffic	
		CAUDHT [12]	BLE	Semi-honest			Gives the pseudonym of infected person of used in encounter	(4)	Uses cover traffic	DHT instead of central server
		Pronto-C2 [6]	BLE	Semi-honest			Gives the pseudonym of infected person of used in encounter		x	Can be implemented with blockchain
		TraceSecure (messaging approach) [7]	BLE	Semi-honest	x	partially	Gives time period of encounter		Uses cover traffic	Relies on HA and government not to cooperate.
		ConTra Corona [10]	BLE	Trusted		x	Gives time period of encounter			Relies on HA and matching server not to cooperate.

REFERENCES

[1] CVE Details. 2019. Vulnerability Details: CVE-2019-2102. www.cvedetails.com/cve/CVE-2019-2102. Accessed: 05. May 2020.

[2] Deutsche Telekom AG and SAP SE. 2020. Corona-Warn-App. www.github.com/corona-warn-app/cwa-documentation. Accessed: 13. May 2020.

[3] Aargauer Zeitung. 2020. Kommission will keine Pflicht für Nutzung von Contact-Tracing-App. www.aargauerzeitung.ch/schweiz/kommission-will-keine-pflicht-fuer-nutzung-von-contact-tracing-app-137710182. Accessed: 26.03.2020.

[4] Thamer Altuwaiyan, Mohammad Hadian, and Xiaohui Liang. 2018. EPIC: Efficient Privacy-Preserving Contact Tracing for Infection Detection. In *ICC*. IEEE, 1–6.

[5] Aradhana Aravindan and Sankalp Phartiyal. 2020. Bluetooth phone apps for tracking COVID-19 show modest early results. www.reuters.com/article/us-health-coronavirus-apps/bluetooth-phone-apps-for-tracking-covid-19-show-modest-early-results-idUSKCN2232A0. Accessed: 06. May 2020.

[6] Gennaro Avitabile, Vincenzo Botta, Vincenzo Iovino, and Ivan Visconti. 2020. Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System. Cryptology ePrint Archive, Report 2020/493.

[7] James Bell, David Butler, Chris Hicks, and Jon Crowcroft. 2020. TraceSecure: Towards Privacy Preserving Contact Tracing. *CoRR* abs/2004.04059 (2020).

[8] Alex Berke, Michiel A. Bakker, Praneeth Vepakomma, Ramesh Raskar, Kent Larson, and Alex 'Sandy' Pentland. 2020. Assessing Disease Exposure Risk With Location Histories And Protecting Privacy: A Cryptographic Approach In Response To A Global Pandemic. *CoRR* abs/2003.14412 (2020).

[9] Stefano Bertuletti, Andrea Cereatti, Ugo Della Della, Michele Caldara, and Michael Galizzi. 2016. Indoor distance estimated from Bluetooth Low Energy signal strength: Comparison of regression models. In *SAS*. IEEE, 1–5.

[10] Wasilij Beskorovajnov, Felix Dörre, Gunnar Hartung, Alexander Koch, Jörn Müller-Quade, and Thorsten Strufe. 2020. ConTra Corona: Contact Tracing against the Coronavirus by Bridging the Centralized–Decentralized Divide for Stronger Privacy. Cryptology ePrint Archive, Report 2020/505.

[11] Bluetooth SIG, Inc. 2020. 2020 Bluetooth Market Update. www.bluetooth.com/bluetooth-resources/2020-bmu/ Last access: 28. April 2020.

[12] Samuel Brack, Leonie Reichert, and Björn Scheuermann. 2020. Decentralized Contact Tracing Using a DHT and Blind Signatures. Cryptology ePrint Archive, Report 2020/398.

[13] Bundesministerium für Justiz und Verbraucherschutz. 2020. Verordnung über die Ausdehnung der Meldepflicht nach § 6 Absatz 1 Satz 1 Nummer 1 und § 7 Absatz 1 Satz 1 des Infektionsschutzgesetzes auf Infektionen mit dem erstmals im Dezember 2019 in Wuhan/Volksrepublik China aufgetretenen neuartigen Coronavirus ("2019-nCoV") § 1 Ausdehnung der Meldepflicht. www.gesetze-im-internet.de/coronavmeldev. Accessed: 11. May 2020.

[14] Matt Burgess. 2020. Coronavirus contact tracing apps were meant to save us. They won't. www.wired.co.uk/article/contact-tracing-apps-coronavirus. Accessed: 30.04.2020.

[15] Justin Chan et al. 2020. PACT: Privacy Sensitive Protocols and Mechanisms for Mobile Contact Tracing. *CoRR* abs/2004.03544 (2020).

[16] Hao Chen, Kim Laine, and Peter Rindal. 2017. Fast Private Set Intersection from Homomorphic Encryption. In *CCS*. ACM, 1243–1255.

[17] Hyunghoon Cho, Daphne Ippolito, and Yun William Yu. 2020. Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-offs. *CoRR* abs/2003.11511 (2020).

[18] Covid Watch. 2020. Covid Watch. www.covid-watch.org Last access: 07. April 2020.

[19] Cristina Criddle and Leo Kelion. 2020. Coronavirus contact-tracing: World split between two types of app. www.bbc.com/news/technology-52355028. Accessed: 11. May 2020.

[20] Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. 2012. Fast and Private Computation of Cardinality of Set Intersection and Union. In *CANS*, Vol. 7712. Springer, 218–231.

[21] George Danezis, Roger Dingledine, and Nick Mathewson. 2003. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *SP*. IEEE Computer Society, 2–15.

[22] Didem Demirag and Erman Ayday. 2020. Tracking and Controlling the Spread of a Virus in a Privacy-Preserving Way. *CoRR* abs/2003.13073 (2020).

[23] Deutsche Welle. 2020. Coronavirus tracking apps: How are countries monitoring infections? www.dw.com/en/coronavirus-tracking-apps-how-are-countries-monitoring-infections/a-53254234 Last access: 30. April 2020.

[24] Deutschlandfunk. 2020. Bundesjustizministerin: Handy-Tracking geht "nur mit Freiwilligkeit". www.deutschlandfunk.de/corona-pandemie-bundesjustizministerin-handy-tracking-geht.694.de.html?dram:article_id=473683. Accessed: 26.03.2020.

[25] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. 2004. Tor: The Second-Generation Onion Router. In *USENIX*. USENIX, 303–320.

[26] Brian Dolan. 2009. SIG Introduces Bluetooth Low Energy Wireless Technology, the Next Generation of Bluetooth Wireless Technology. www.mobihealthnews.com/5828/sig-introduces-bluetooth-low-energy-wireless-technology-the-next-generation-of-bluetooth-wireless-technology. Accessed: 05. May 2020.

[27] Ken TD Eames and Matt J Keeling. 2003. Contact tracing and disease control. *P ROY SOC B-BIOL SCI* 270, 1533 (2003), 2565–2571.

[28] Lilian Edwards, Michael Veale, Orla Lynskey, Carly Kind, and Rachel Coldicutt. 2020. The Coronavirus (Safeguards) Bill 2020: Proposed protections for digital interventions and in relation to immunity certificates. LawArXiv. www.osf.io/preprints/lawarxiv/yc6xu.

[29] Ramsey Faragher and Robert Harle. 2014. An analysis of the accuracy of Bluetooth low energy for indoor positioning applications. In *ION GNSS+*, Vol. 812. 201–210.

[30] Ramsey Faragher and Robert Harle. 2015. Location Fingerprinting With Bluetooth Low Energy Beacons. *IEEE J. Sel. Areas Commun.* 33, 11 (2015), 2418–2428.

[31] Luca Ferretti et al. 2020. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science* (2020).

[32] FluPhone Study Team. 2011. FluPhone Project: Understanding Spread of Infectious Disease and Behavioural Responses. www.cl.cam.ac.uk/research/srg/netos/projects/archive/fluphone2/. Accessed: 04. May 2020.

[33] Matheus E Garbelini, Sudipta Chattopadhyay, and Chundong Wang. 2020. SweynTooth: Unleashing Mayhem over Bluetooth Low Energy. www.asset-group.github.io/disclosures/sweyntooth/sweyntooth.pdf. Accessed: 05. May 2020.

[34] Oded Goldreich, Silvio Micali, and Avi Wigderson. 1987. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In *STOC*. ACM, 218–229.

[35] Google and Apple. 2020. Exposure Notification - Bluetooth Specification. www.covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-BluetoothSpecificationv1.2.pdf Last access: 18. May 2020.

[36] Google and Apple. 2020. Privacy-Preserving Contact Tracing. www.apple.com/covid19/contacttracing Last access: 24. April 2020.

[37] Government of Singapore. 2020. BlueTrace. www.bluetrace.io Last access: 26. April 2020.

[38] Government of Singapore. 2020. TraceTogether. www.tracetogether.gov.sg Last access: 06. April 2020.

[39] Government of Singapore - Ministry of Health. 2020. Two Charged Under Infectious Diseases Act for False Information and Obstruction of Contact Tracing. www.moh.gov.sg/news-highlights/details/two-charged-under-infectious-diseases-act-for-false-information-and-obstruction-of-contact-tracing Last access: 26. April 2020.

[40] Yaron Gvili. 2020. Security Analysis of the COVID-19 Contact Tracing Specifications by Apple Inc. and Google Inc. Cryptology ePrint Archive, Report 2020/428.

[41] Josef Hallberg, Marcus Nilsson, and Kare Synnes. 2003. Positioning with bluetooth. In *ICT*, Vol. 2. IEEE, 954–958.

[42] Isobel A Hamilton. 2020. 11 countries are now using people's phones to track the coronavirus pandemic, and it heralds a massive increase in surveillance. www.businessinsider.com/countries-tracking-citizens-phones-coronavirus-2020-3?r=DE&IR=T. Accessed: 26.03.2020.

[43] Arvin Hekmati, Gowri Sankar Ramachandran, and Bhaskar Krishnamachari. 2020. CONTAIN: Privacy-oriented Contact Tracing Protocols for Epidemics. *CoRR* abs/2004.05251 (2020).

[44] Ramon Huerta and Lev S Tsimring. 2002. Contact tracing and epidemics control in social networks. *PHYS REV E* 66, 5 (2002), 056115.

[45] Institut national de recherche en informatique et en automatique (INRIA). [n.d.]. ROBust and privacy-presERving proximity Tracing protocol. www.github.com/ROBERT-proximity-tracing/documents Last access: 26. April 2020.

[46] Andrea Jelinek. [n.d.]. Letter from the European Data Protection Board to the European Commission. edpb.europa.eu/sites/edpb/files/files/file1/edpbletterecadvisecodiv-appguidance_final.pdf Last access: 15. May 2020.

[47] Seungyeon Jeong, Seungho Kuk, and Hyogon Kim. 2019. A Smartphone Magnetometer-Based Diagnostic Test for Automatic Contact Tracing in Infectious Disease Epidemics. *IEEE Access* 7 (2019), 20734–20747.

[48] Malek Karaim, Mohamed Elsheikh, and Aboelmagd Noureldin. 2018. *GNSS Error Sources*. IntechOpen. https://doi.org/10.5772/intechopen.75493

[49] Ágnes Kiss, Jian Liu, Thomas Schneider, N. Asokan, and Benny Pinkas. 2017. Private Set Intersection for Unequal Set Sizes with Mobile Applications. *PoPETs* 2017, 4 (2017), 177–197.

[50] Antti Kotanen, Marko Hännikäinen, Helena Leppäkoski, and Timo Hämäläinen. 2003. Experiments on Local Positioning with Bluetooth. In *ITCC*. IEEE Computer Society, 297–303.

[51] James Larus et al. 2020. Joint Statement on Contact Tracing: Date 19th April 2020. www.drive.google.com/file/d/1OQg2dxPu-x-RZzETlpV3lFa259Nrpk1J/view. Accessed: 30.04.2020.

[52] Hui Liu, Houshang Darabi, Pat P. Banerjee, and Jing Liu. 2007. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE T SYST MAN CY C* 37, 6 (2007), 1067–1080.

[53] Joseph K. Liu et al. 2020. Privacy-Preserving COVID-19 Contact Tracing App: A Zero-Knowledge Proof Approach. Cryptology ePrint Archive, Report 2020/528.

[54] Shu Liu, Yingxin Jiang, and Aaron Striegel. 2013. Face-to-face proximity estimation using bluetooth on smartphones. *IEEE T MOBILE COMPUT* 13, 4 (2013), 811–823.

[55] Shu Liu and Aaron Striegel. 2011. Accurate Extraction of Face-to-Face Proximity Using Smartphones and Bluetooth. In *ICCCN*. IEEE, 1–5.

[56] Massachusetts Institute of Technology. 2020. Private Kit: Safe Paths; Privacy-by-Design Contact Tracing. www.safepaths.mit.edu Last access: 06. April 2020.

[57] Alessandro Montanari. 2015. Multimodal Indoor Social Interaction Sensing and Real-time Feedback for Behavioural Intervention. In *S3@MobiCom*. ACM, 7–9.

[58] Michael Naehrig, Kristin E. Lauter, and Vinod Vaikuntanathan. 2011. Can homomorphic encryption be practical?. In *CCSW*. ACM, 113–124.

[59] Futoshi Naya, Haruo Noma, Ren Ohmura, and Kiyoshi Kogure. 2005. Bluetooth-based Indoor Proximity Sensing for Nursing Context Awareness. In *ISWC*. IEEE Computer Society, 212–213.

[60] Khuong An Nguyen, Chris Watkins, and Zhiyuan Luo. 2017. Co-location epidemic tracking on London public transports using low power mobile magnetometer. In *IPIN*. IEEE, 1–8.

[61] Rishab Nithyanand, Oleksii Starov, Phillipa Gill, Adva Zair, and Michael Schapira. 2016. Measuring and Mitigating AS-level Adversaries Against Tor. In *NDSS*. The Internet Society.

[62] PePP-PT e.V. i.Gr. 2020. PePP-PT. www.pepp-pt.org Last access: 05. April 2020.
[63] PePP-PT e.V. i.Gr. 2020. PePP-PT Documentation. www.github.com/pepp-pt/pepp-pt-documentation Last access: 26. April 2020.
[64] Krzysztof Pietrzak. 2020. Delayed Authentication: Preventing Replay and Relay Attacks in Private Contact Tracing. Cryptology ePrint Archive, Report 2020/418.
[65] Benny Pinkas and Eyal Ronen. 2020. Hashomer Crypto Reference. github.com/eyalr0/HashomerCryptoRef. Accessed: 12. May 2020.
[66] Aswin N. Raghavan, Harini Ananthapadmanaban, Manimaran Sivasamy Sivamurugan, and Balaraman Ravindran. 2010. Accurate mobile robot localization in indoor environments using bluetooth. In *ICRA*. IEEE, 4391–4396.
[67] Leonie Reichert, Samuel Brack, and Björn Scheuermann. 2020. Privacy-Preserving Contact Tracing of COVID-19 Patients. Poster Session at the 41st IEEE Symposium on Security and Privacy.
[68] Mohamed Rida, Fuqiang Liu, Yassine Jadi, Amgad Abdullah, and Ahmed Askourih. 2015. Indoor Location Position Based on Bluetooth Signal Strength. In *ICISCE*. 769–773.
[69] Ronald L. Rivest et al. 2020. The PACT protocol specification. pact.mit.edu/. Accessed: 13. May 2020.
[70] Miguel Rodriguez, Juan P Pece, and Carlos J Escudero. 2005. In-building location using bluetooth. In *IWWAN*.
[71] James Scott, Pan Hui, Jon Crowcroft, and Christophe Diot. 2006. Haggle: A Networking Architecture Designed Around Mobile Users. In *WONS*.
[72] Hyonhee Shin and Josh Smith. 2020. South Korea scrambles to contain nightclub coronavirus outbreak. www.reuters.com/article/us-health-coronavirus-southkorea/south-korea-scrambles-to-contain-nightclub-coronavirus-outbreak-idUSKBN22N0DA. Accessed: 11. May 2020.
[73] Selena Simmons-Duffin and Robert Stein. 2020. CDC Director: 'Very Aggressive' Contact Tracing Needed For U.S. To Return To Normal. www.npr.org/sections/health-shots/2020/04/10/831200054/cdc-director-very-aggressive-contact-tracing-needed-for-u-s-to-return-to-normal?t=1588258166718&t=1588258706121. Accessed: 26.03.2020.
[74] Natasha Singer and Choe Sang-Hun. 2020. As Coronavirus Surveillance Escalates, Personal Privacy Plummets. www.nytimes.com/2020/03/23/technology/coronavirus-surveillance-tracking-privacy.html. Accessed: 26.03.2020.
[75] Nigel P. Smart. 2016. *Cryptography Made Simple*. Springer.
[76] The MITRE Corporation. 2020. CVE-2020-0022. www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-0022. Accessed: 02. June 2020.
[77] The New York Times. 2020. Lockdowns in France and U.K. Expected to Last Into Next Month. www.nytimes.com/2020/04/13/world/coronavirus-news-world-international-global.html Last access: 04. May 2020.
[78] The New York Times. 2020. To Track Coronavirus, Israel Moves to Tap Secret Trove of Cellphone Data. www.nytimes.com/2020/03/16/world/middleeast/israel-coronavirus-cellphone-tracking.html?referringSource=articleShare Last access: 06. April 2020.
[79] Ni Trieu, Kareem Shehata, Prateek Saxena, Reza Shokri, and Dawn Song. 2020. Epione: Lightweight Contact Tracing with Strong Privacy. *CoRR* abs/2004.13293 (2020).
[80] Carmela Troncoso et al. 2020. Decentralized Privacy-Preserving Proximity Tracing. www.github.com/DP-3T/documents Last access: 28. May 2020.
[81] Carmela Troncoso et al. 2020. Decentralized Privacy-Preserving Proximity Tracing - Version: 25 May 2020. www.github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf Last access: 28. May 2020.
[82] Serge Vaudenay. 2020. Analysis of DP3T. Cryptology ePrint Archive, Report 2020/399.
[83] Serge Vaudenay. 2020. Centralized or Decentralized? The Contact Tracing Dilemma. Cryptology ePrint Archive, Report 2020/531.
[84] Glenn Webb, Cameron Browne, Xi Huo, Ousmane Seydi, Moussa Seydi, and Pierre Magal. 2015. A model of the 2014 Ebola epidemic in West Africa with contact tracing. *PLoS currents* 7 (2015).
[85] Eiko Yoneki. 2011. FluPhone study: virtual disease spread using haggle. In *CHANTS@MobiCom*. ACM, 65–66.
[86] Sheng Zhou and John K. Pollard. 2006. Position measurement using Bluetooth. *IEEE T CONSUM ELECTR* 52, 2 (2006), 555–558.