A Survey of Automatic Contact Tracing Approaches Using Bluetooth Low Energy

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. Bluetooth Low Energy (BLE) as base technology has the most promising properties, so this survey focuses on automated contact tracing techniques using BLE. We define multiple classes of methods and 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 Using Bluetooth Low Energy. 1, 1 (October 2020), 29 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 [15, 25, 52, 106]. To mitigate its spreading, social and economic life was shut down in affected areas [105]. 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 [38, 99]. 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 [52, 100, 106]. There are also concerns about an increase of discrimination of socio-economic or ethnic groups through the adoption of automatic contact tracing (ACT) [62]. In many countries, nationwide adoption of ACT applications cannot be enforced by the state [2, 26, 35, 58]. To ensure great effectiveness it is therefore essential that citizens trust the ACT system enough to participate voluntarily. System designs that send detailed location or contact histories to a government-run central

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.

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.

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

The goal of this survey is to provide a general overview of different types of approaches for ACT with a focus on privacy. As the majority of real-world ACT applications are based on Bluetooth Low Energy, especially those with user privacy as a design goal, we will concentrate on approaches using this technology. Some notable examples for system designs that utilize other tracing methodologies are also included. We identify two larger groups and several subgroups of architectures for ACT. We discuss shortcomings of each subgroup and problems common to all contact tracing systems based on this technology.

In the following 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. Central servers are mostly used for relaying messages in these approaches. Section 5 deals with common security issues and threats but also talks about additional features of ACT systems. 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 [34, 113]. Both in theory and in practice it has proven to be a useful tool for containing epidemics. Stochastic modeling was used in [34, 55, 113] 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 [34]. 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 difficult for airborne diseases like SARS, MERS or Covid-19 [34]. This is due to the fact that random encounters are complicated to notify as the diagnosed person can then oftentimes not provide information about all relevant contacts.

2.2 Automated Contact Tracing

To ensure warnings are delivered fast to users at risk 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 [38]. In many countries smartphone apps are discussed for this purpose. These inform users of past close 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 [40]. The goal was to better understand and predict the influenza epidemic and how people alter their behaviour in response. As part of the project a field trial was conducted [114], in which participants downloaded an app onto their phone that checked for other devices in proximity using Bluetooth. For detecting phones close by, the FluPhone project built upon Haggle [97], 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 infection. The system also had the capability of marking devices as infected which could subsequently contaminate other users' devices they encountered based on probability calculations.

Research in the field of ACT has been slow but steady [3, 39, 59, 66, 79, 88, 95, 96, 115, 116]. Besides the flu, other epidemics such as SARS, the swine flu, MERS, Ebola, H1N1 and Zika moved into focus. With the 2020 Covid-19 pandemic, many new approaches were proposed and implemented during the first half of the year. The first country to roll out a full ACT application for Covid-19 was Singapore with TraceTogether [47].

For ACT to be successful, a wide spread adoption is needed. Simulations evaluating the effectiveness of ACT use adoption rates from 40% [81] and 53% [65] to 56% [54] of the population. It has been suggested by some authors that these numbers should not be understood as hard limits, as apps do not become useless at lower adoption rates but rather less effective [82]. In another study Kretzschmar et. al. [64] find that ACT is more effective than manual contact tracing, even if only 20% of the population use the tracing app. This effectiveness derives from the shorter delay of notifying contacts compared to the manual approach of interviewing a patient and then calling their previous contacts via telephone.

One factor impacting adoption is public perception. Systems perceived as surveillance are seen as less trustworthy and people are less inclined to install the corresponding ACT app on their devices [13]. Some states, such as China, have therefore made the use of the local ACT app a requirement for using public transit and participating in public life [23]. For most countries, this is not an option as such measures spark serious concerns regarding civil liberties and discriminate against people without a modern smartphone [2, 26, 58].

Usability also has to be taken into account when talking about adoption rates of ACT systems. A study by Trang et al. [107] shows that users who are educated about the benefits for themselves and for society are more likely to adopt it. Due to many apps being voluntary to use, the study finds that an intrinsic motivation for using ACT apps is an important factor for its penetration. Another usability requirement often formulated is that the ACT app should not drain too much power from the smartphone as people might uninstall it for this reason [46]. Also users should not be disturbed by the application and it should therefore be capable of running in the background without the need to open the app regularly [46]. A similar requirement is automatic processing of data without user interaction, as well as refraining from having users perform manual tasks that are error-prone and time-consuming such as entering a long number read out over telephone [33]. Usability requirements from the health authority's (HA) side are that an ACT system should allow them to automate as many tasks as possible or delegate them towards medical staff conducting tests [33]. For this purpose redundant and various communication channels from the HA to the app users are necessary so that additional information can be spread effectively and quickly.

It has been discouraged to consider ACT as replacement for manual contact tracing [46, 90]. ACT systems might be faster, more scalable and once installed less costly. But the manual approach has been proven to be effective in past epidemics, is already in place and provides rich human-to-human interaction. Human contact tracers are also capable of detecting non-direct methods of transmission through questions. For this purpose they require an infected user's location history to trace potential contacts. Some apps are specifically designed to support manual contact tracing processes [16, 46, 93, 95, 103]. ACT systems deployed to combat Covid-19 are generally used in combination with existing procedures.

One hope connected to the deployment of ACT systems has often been that lockdowns will be shortened and life will return back to normal [8]. But what means are acceptable to achieve this goal? Some states use the pandemic to employ systems that might later prove to be capable of dual-use [52, 62, 100, 106]. In opposition, many new ideas how

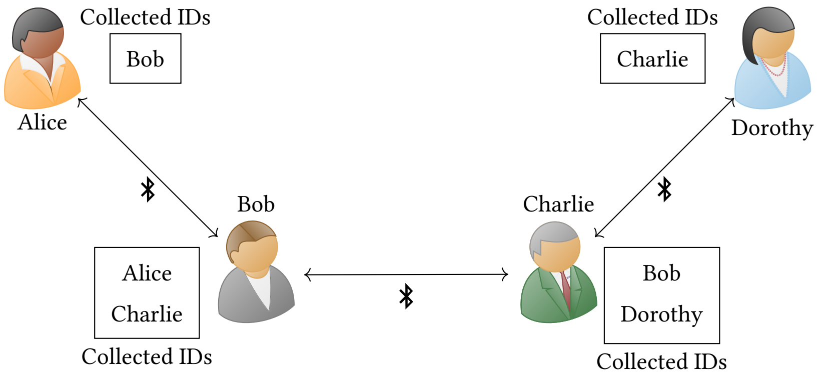

Fig. 1. During contact collection, each user stores the IDs of all devices that are in proximity. These IDs can be used to notify close contacts in case of a subsequently detected infection.

ACT systems can be designed without becoming part of the surveillance infrastructure have been proposed and rolled out. This survey discusses a variety of these ideas.

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 [51, 63, 78, 94, 117]. These works use the receiver signal strength indicator (RSSI) to measure distance between receiver and transmitter and thereby derive a location. Raghavan et al. [89] were able to show that Bluetooth version 2.0 can be used for localization with an error of less than 45cm. Liu et al. [70, 71] demonstrated that Bluetooth is efficient for detecting face-to-face interactions by providing a model for estimating distance using RSSI readings. Bluetooth has the problem that it is an active protocol where a connection is established between the two parties before any payload can be exchanged. This potentially hinders an effective exchange of messages due to the added complexity of the connection establishment. Additionally, since devices advertise themselves, they signal to possible attackers where to find an activated interface, which can then be exploited using known vulnerabilities such as [104].

Bluetooth specification 4.0 introduced Bluetooth Low Energy (BLE), an energy-efficient, short-range variant of standard Bluetooth [28]. In 2020, both Bluetooth and BLE have a high adoption rate, as 100% of new smartphones support both standards [11]. Due to its battery saving properties BLE was adapted for positioning and proximity detection [36, 37, 75, 92]. Bertuletti et al. [9] were able to reduce the error of BLE-based location measurements to less than 40cm. BLE has therefore been adopted by Singapore's TraceTogether to realize ACT. Here, users continuously send out pseudonyms over BLE as shown in Figure 1. These beacons can be received and recorded by other users. If a person is diagnosed, the pseudonyms they have seen in the past are used to identify their random encounters. One shortcoming of BLE (and Bluetooth) for proximity detection and contact tracing is the large variability of transmission power over different smartphone types. RSSI readings have to be calibrated to the respective devices [46]. For usability reasons it is also essential that an ACT application using these technologies can run in the background. Apple's iOS restricts the usage of the corresponding interfaces for apps running in the background, thereby interfering with co-location detection [46]. This limitation concerns all types of sensors except those related with location tracking. BLE has vulnerabilities that make it exploitable to attackers when turned on [1, 41].

Bluetooth and BLE are not the only technologies for determining co-location. Methods like GPS [74, 91], cell tower triangulation [106], Wifi [3], or correlating Magnetometer readings [59, 79] could also be used for ACT. But all of these technologies have shortcomings which we will discuss in the following. GPS data is generally seen as very privacy 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 or inside [60]. 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 kinds of false positive errors are difficult to mitigate when using GPS or cell tower triangulation. Both technologies are too imprecise to derive meaningful data about interactions of users. Wifi, just like Bluetooth/BLE, has the advantage of being blocked by objects such as walls.

While Wifi has been widely used for indoor positioning [68], just like cell tower triangulation it requires infrastructure that might not be available everywhere, especially outdoors or in remote locations. It is therefore not suitable for ACT, which is required to function anywhere.

Correlating magnetometer reading of users is another passive method suitable for ACT. It requires little energy while working indoors and outdoors. When two magnetometer readings have a similar variance during the same time period this indicates that they were recorded at the same location. No information about the distance between the people recording these traces can be deduced. But proximity information is crucial for evaluating the likelihood of transmissions in an ACT setting [102]. There has been little research in the area of co-location detection using magnetometers so far and it is not as well investigated as BLE. So while this method works in the laboratory, reproducing the findings on large scale might be difficult, making this technology inadequate for ACT as timely deployment is vital.

The Fluphone project tested RFID tags [40] to detect co-location. While this approach is interesting, tags need to be distributed to all users. This overhead is considerably larger than providing an app in various app stores and using common smartphone capabilities. Since the ID of RFID tags is static, this technology also allows the re-identification of users, making it easy to track their location.

As we have seen, BLE as is the most suitable base-technology for ACT. For this reason, most systems proposed and deployed are built on this approach to detect contacts between users. The main differences between various approaches to ACT lie in the way how risk assessment is conducted and which parties hold relevant data. In the remainder of this survey, we will therefore focus on works using BLE for co-location detection. Notable approaches, which use other sensors but can be realised using BLE, are also discussed.

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, 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): This is 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. Another possible goal is the deanonymization of users at risk. Since infections with SARS-CoV-2, the virus causing Covid-19, have to be reported in many countries [14, 48], 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 [48]. 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 who is infected or who infected them. It is hereby important to distinguish who an attacker focuses on: random users, close social contacts who are regularly in presence of the attacker, or public figures which are easy to track down by a *Curious Stalker* or *Paparazzi*. The stalker can follow victims and observe if their habits change.
(3) Infected users: 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 [98] or other forms of social outcasting. But infected users can also be malicious, by trying to figure out who they have infected.
(4) Eavesdroppers: Eavesdroppers are passive attackers that listen to communication in the protocol, both on the wireless network as well as the communication with a centralised backend. Users, network, and service operators can all take the role of an eavesdropper in a protocol.
(5) Service operators: 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.
(6) 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.

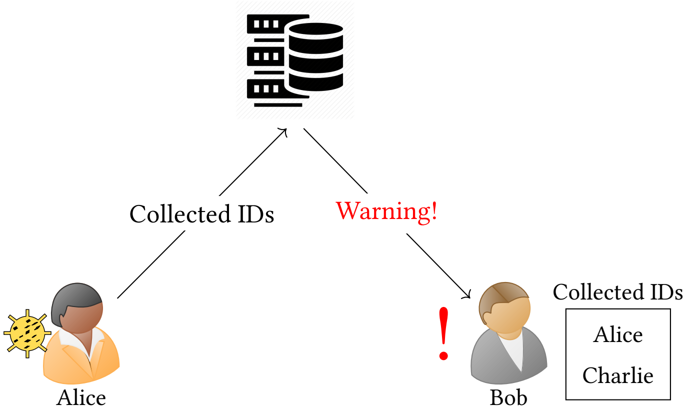

Fig. 2. Here the general idea of server based ACT is visualized. Alice sends her collected IDs to the server when diagnosed as infected. The server does a risk assessment for her contacts and warns Bob.

3 SERVER-SIDE RISK ASSESSMENT

There are numerous ways 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 [47]. The app is called TraceTogether, while the associated open source project has the name Bluetrace [46]. End devices of users run an application which uses ephemeral BLE beacons to advertise their presence. These pseudonyms are generated on the central server, so that the server always knows 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 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 (see Figure 2). 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. If a certain risk threshold is exceeded, the server will notify the corresponding 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 BlueTrace can be found in the framework of PePP-PT [83]. PePP-PT is a European initiative also focusing on a centralized approach. Similar to BlueTrace, the central server is operated by a country's health authority. Pseudonyms for BLE are generated by the server and sent to the user's device which announces them over its Bluetooth interface. These pseudonyms are encrypted values of the user's 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 users at risk can be calculated. Users at risk are then notified with push notifications.

Two implementations of PePP-PT exist; PePP-PT NTK [84] and ROBERT [57]. They only differ in minor details. For instance, ROBERT uses 3DES as their symmetric encryption algorithm instead of AES. An app based on ROBERT has been released in France under the name of Stop Covid [56]. To facilitate cooperation between different states, both BlueTrace and PePP-PT allow for cooperation between different health authorities.

The described models for server-based ACT 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 users at risk. Recipients 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 is leaked to the entity operating the servers, which is either the HA or a service operator. 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 server 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 specific details about the nature of users' relations can be revealed. 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 BLE traces can be deanoymized by any eavesdropper on the BLE band. It has been proposed to use rotating keys to reduce this threat [112]. An attacker observing the network does not learn who is at risk. But uploads to the server will reveal who is infected if no additional measures such as cover traffic or hiding the IP address are taken.

Other systems with even less focus on user privacy exist as well. The Indian government developed an app called Aarogya Setu [77] to mitigate the spread of Covid-19. Essential parts of the app and infrastructure are not open-source and it is unclear how exactly a risk score is determined. An analysis of the app found that GPS and BLE are used [49]. Infection chains are recorded by sending and receiving fixed pseudonym IDs that are uploaded to the HA in case of an infection, as well as the location history of the last 30 days. This allows the HA to not only notify individuals about infection risks but also to detect events and conditions where multiple infections took place.

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 [4], 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 [4, 15, 67]. These two aspects have greatly influenced the public discussion in some European countries causing governments to move away from centralized approaches as described in this section [20].

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 querying users themselves. These approaches leverage modern cryptographic tools such as homomorphic encryption and secure multiparty computation.

3.2.1 Homomorphic Encryption. Homomorphic encryption (HE) [76] 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. 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 [3]. Here, users do not actively send out BLE 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 own 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 requesting user who can decrypt it and derive their personal risk score. Users do not learn the location traces of individual infected users, but will learn at which locations they have been close to an infected person. They can also forge their upload to verify assumptions about the risk status of a person. Infected users can provide fake locations to drive businesses away from certain shops or shame targets. Service and network operators are not able to learn the locations, risk scores or health status of healthy users because data is encrypted with a secure key belonging to the user. To not reveal a pattern, users at risk do have to continue querying the server after being diagnosed.

Another approach using homomorphic encryption was proposed by Bell et al. [6]. We call this approach "HE-based TraceSecure". The system relies on a pseudonyms exchange over BLE. It reveals to the server (which is run by the HA or a service operator) who has interacted with whom but keeps the health status secret from the server and non-colluding network operators. The authors see this leakage of interactions as a feature that can be used for building a social graph of pseudonyms. This graph can then be used as part of a privacy-preserving evaluation of social distancing policies. Users of HE-based TraceSecure learn which pseudonym infected them.

3.2.2 Secure Multiparty Computation. The field of *secure multi-party computation* (MPC) [101, Chapter 22] deals with creating protocols for joint computation on private, distributed data. It studies mechanisms to allow a group

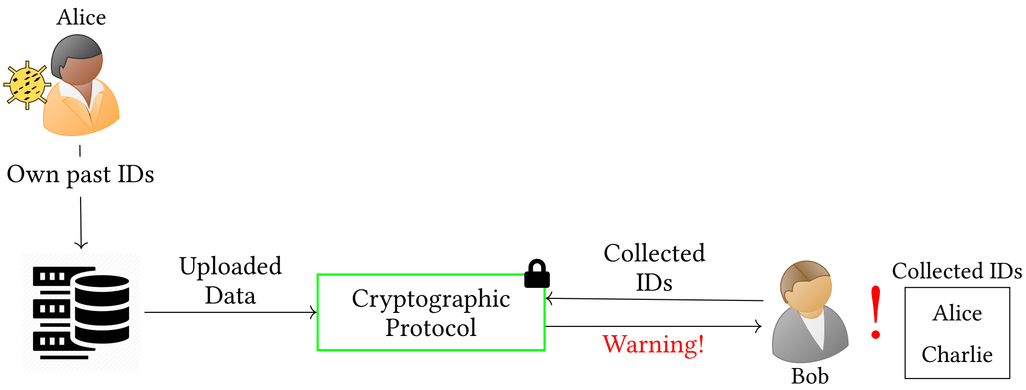

Fig. 3. This figure illustrates how a simple private set intersection protocol for BLE-based ACT could work. This example does not leak the intersection to Bob.

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 [101, Chapter 22.2]. For ACT, generally only two parties are considered, a server and a client who wants to determine their risk status. One way of realizing arbitrary MPC protocol are *Yao's garbled circuits* [101]. 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 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 one of these using an index without learning the input of their counterpart.

A useful application of MPC is private set intersection (PSI). Two participants each hold a set of elements and want to calculate their 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. [7] use Diffie-Hellman PSI for ACT. Instead of exchanging BLE 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. The recipient then encrypts 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 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. The server, and therefore the HA or a service operator, does not learn which data was provided by a user. Users does not learn where infected people have been. Network operators will not know who is at risk or diagnosed as long as infected user continue querying if they have been diagnosed. The client learns if they are at risk and since the intersection is leaked they will also know where the encounter occurred. Since GPS data is used, malicious users can forge input and for example provide the home address of a target. Infected users can provide wrong locations to make certain places look bad and scare customers of local businesses.

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 [91]. Risks and attack vectors are the same as for Berke et al.'s solution.

The protocol of Demirag et al. [24] uses Bluetooth/BLE to advertise a static pseudonym. The authors do not go into detail if their system relies on regular Bluetooth or BLE. The HA server holds all pseudonyms of people with verified infections. To figure out how many people they have met in the last weeks that were infected, a user performs PSI with the server following the protocol of De Cristofaro et al. [21]. This protocol only provides the size of the intersection. Similarly to the two protocols discussed above, this system requires the central server to know relevant information about the infected individuals, here the pseudonyms they have used in the past. The server (i. e., the HA or a service operator) does not learn which pseudonyms the client used as input for the set intersection or if they are at risk. The client does not learn the pseudonyms of users that are infected, not even the ones they have been in contact with. If users continue communicating with the system even after receiving a positive diagnose, a network operator can not tell if they are at risk or even infected.

Epione proposed by Trieu et al. [108] also uses Bluetooth technology to exchange pseudonyms. For each encounter both parties create a new pseudonym. They use a 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. In Epione, the HA and the central server are required to know the pseudonyms infected users have used in the past. The server, meaning the HA or a service operator, does not learn the past pseudonyms of other users or who is at risk. The client only learns how many risky encounters they have had but not the corresponding pseudonyms of infected users. An eavesdropper on the network will not be able to distinguish traffic from users at risk from normal traffic, but to eliminate the possibility of leaking data through patterns, diagnosed users have to continue using the system.

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 [42]. 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.

Runtime and communication overhead remain problematic in designs relying on the described cryptographic building blocks. MPC circuits can become very large and may require many gigabytes of data to be communicated. This is hardly feasible on metered mobile data connections. Mobile energy consumption is also limited due to battery sizes. More important even, the general public acceptance of ACT relies on its usability on mobile device. There is research on PSI which attempts to take load off the end devices [61].

Another problem for cryptographic approaches is that DDoS against the central server remains a problem. Due to necessary complex operations executed by the central server, an attacker could aim to exhaust the server's resources with cheaply generated data that is sent to the server.

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. Data passes these infrastructures, but no

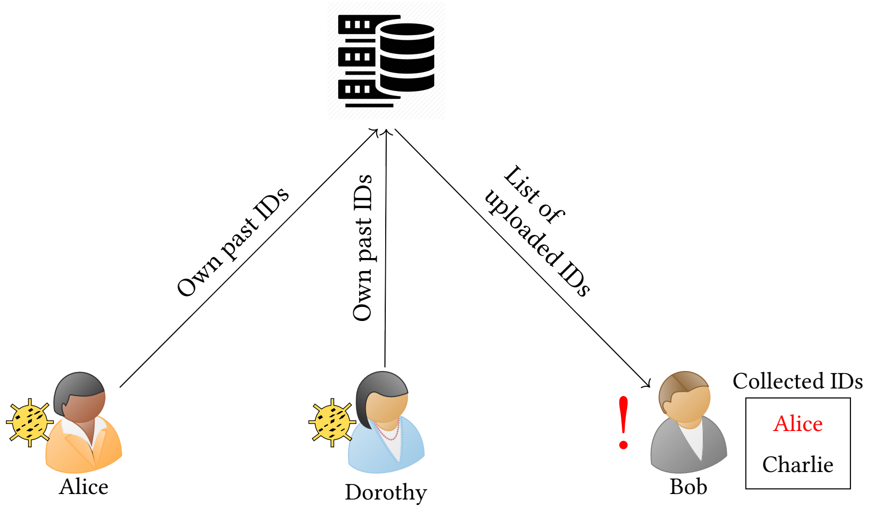

Fig. 4. This figure visualizes the idea of a simple broadcast ACT model. When Alice is verified as infected, she will upload the IDs she advertised in the past to a server. Bob will download a list from the server containing Alice's IDs, but also those of other infected users such as Dorothy. Checking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

information about social interactions is revealed and either infected users and/or the users at risk remain private. This technique often requires more resources on end devices. This type of ACT is often also called *decentralized*. Several models using client-side risk assessment are discussed in this section. We distinguish between systems using broadcasts and ones using direct messaging. A simplified illustration of the first idea is displayed in Figure 4.

4.1 Broadcast Models

DP-3T [109] is an initiative using the broadcast approach. In DP-3T's early so-called low-cost design users use an individual seed to derive a daily key. This daily key is then used to deterministically calculate rotating BLE pseudonyms. 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 application locally derives the corresponding pseudonyms 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 pseudonyms become 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 [110]. Users 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 [17]. When data is uploaded the server, and thereby the HA or the service operator, does learn the past pseudonyms of an infected user but not with whom they interacted.

Apple and Google, two companies together dominating the market for smartphone operating systems, formed an alliance to present a joint approach for ACT (which we will call "GAEN") [44]. They propose a technical specification for an API after DP-3T and other similar schemes like CONTAIN [53], East-Coast PACT by Rivest et al. [93], and West-Coast PACT by Chan et al. [16] have been broadly discussed. Differences between these schemes and the GAEN framework are mostly on an implementation level. While DP-3T derives the daily key by hashing the key from the day before, GAEN combines the initial tracing key with the number of the day in a key derivation function. Another difference is how pseudonyms 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. GAEN derives pseudonyms 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.

Pinkas and Ronen proposed a similar system called Hashomer relying on an elaborate key derivation mechanism [86]. While the keys advertised at different epochs of the same day are unlinkable, this scheme uses a daily key that can also be published by the central server to reduce the amount of data to be downloaded daily by users. Daily keys from different days are also unlinkable. In Hashomer, the HA only learns past pseudonyms of infected users but not who they interacted with.

Covid-Watch [19] is a project supported by the University of Stanford also employing a broadcast approach. Instead of ephemeral pseudonyms, a new random number is generated per contact event. Another difference to the projects mentioned above is that when a user is tested positive, they will not only upload their own numbers used in the past but also those they recorded. These pairs of data are then broadcast to all other users who check locally if they have a corresponding encounter stored. While Covid-Watch relies on Bluetooth for co-location detection, the authors do not go into detail if the system relies on regular Bluetooth or BLE. When an infected user uploads their history of encounters, the server learns the unique pair of pseudonyms of both sides. If both sides of an encounter upload their history, the server which is run by either the HA or a service operator, can deduce who interacted with whom.

Approaches using the broadcast model are able to hide from the HA the fact that someone has been in contact with a person who was tested positive. 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 encounter 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. This class of attacks requires the attacker to gain access to recent pseudonyms of a victim, which can be obtained by sustaining physical proximity to the targeted victim. In some cases, it is required to get access to the keys that are used for generation. This can only be done by breaking into the victim's phone. A successful untargeted impersonation attack would require the attacker to guess a valid pseudonym which is very unlikely to happen due to the high entropy of randomly generated pseudonyms.

Since risk scoring is done locally in broadcast-based ACT, a network operator will not know who is at risk. But through uploads to the server, network and service operator can learn who is infected.

4.2 Direct Messaging

Another way of doing ACT risk assessment on client devices are postbox systems. The approach was first described by Cho et al. [18] (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 pseudonyms of other users to notify them. To do so, they place a message encrypted with the other

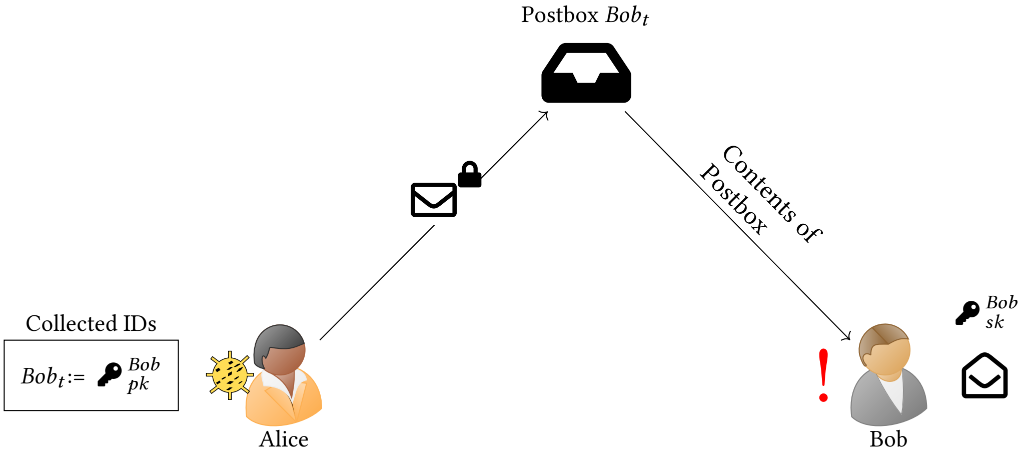

Fig. 5. An example of a direct messaging approach to ACT following Cho et al. [18]. Alice collected Bob's ID Bob_t which she uses to encrypt a message for Bob. This message is placed in the corresponding Postbox where Bob can retrieve it. After its decryption he knows that he is at risk.

user's pseudonym into the corresponding postbox. Users regularly check postboxes belonging to their past advertised pseudonyms to see if a new message has arrived. The message will contain a warning but will not indicate who has sent it. This means the user at risk can not deanonymize infected users. To ensure that the server, and thereby the HA or a service operator, cannot link real identities with postboxes, the authors require requests to the server to be sent through a network of proxies. To mitigate deanonymisation by a network observer which observes traffic it is also necessary to introduce cover traffic. This means users not only send messages to others when they become infected but they also send messages stating that they are still healthy. The server therefore only sees one user placing messages in a postbox but can not decrypt this message and find out if the message is real or a decoy. 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 called CAUDHT, 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 Alice becomes infected, she blinds her past pseudonyms and individually send them to the HA. The HA signs each blinded message it receives and sends it back. It does not learn the plaintext message content during this process. Alice can then unblind the returned values and now holds valid signatures for her pseudonyms. The step of requesting blind signatures has to be additionally secured using, for example, permission keys. 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. To notify an an at-risk contact Bob, Alice places an encrypted message in the postbox corresponding to a pseudonym of Bob she has seen in the past. The message is encrypted using Bob's pseudonym (which is also a public key) and contains the pseudonym Alice advertised at the time of the encounter as well as the HA's signature belonging to this pseudonym. Bob collects the 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. Similarly to Cho et al., cover traffic can be created at random by any user. When a message is passed to the DHT, the network only sees a user placing a message in a postbox, but can not verify if the message is real or a decoy. Since the HA's only task is issuing blind signatures, it does not learn additional information about infected users or who is at risk. In CAUDHT, the pseudonym of the infected user used during the recorded encounter is part of the sent message. This gives the user at risk the option to do a sanity check, but it also means that they might be able using their memory and the time of the encounter to identify the person who infected them.

Another message-based ACT protocol inspired by Cho et al. is message-based TraceSecure [6]. This 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 pseudonyms to the government. In return the user is given a static ID 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. Each message contains an observed pseudonym. Since the government knows the seeds from which pseudonyms are generated, it can derive which static IDs 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, so the messaging service and a network observer do not learn who is infected. Since the HA holds the seeds for all users, it can derive a user's current advertised pseudonym and use this information for tracking. But it does not know who received a warning and is at risk. The government learns the static ID of infected users and people they have been in contact with. The privacy of users relies on the server not being able to link this static IDs to real identities. Users in this system only learn that they are at risk but no additional information and can therefore not conduct meaningful attacks.

ConTra Corona [10] also relies on multiple non-colluding central parties. Ephemeral 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. The first option leaks users identities to the HA and the service operator. This means they will learn who is at risk if no additional measures like hiding the IP addresses are taken. Another attack vector is query timing, which allows the attacker to identify keys belong to the same user. It is also important that a positive and a negative answer will look the same to an eavesdropper on the network. The second option where users do the risk assessment locally does not have these risks. Users in ConTra Corona will only learn during which time period they were infected but not by which pseudonym.

A system comparable in concept to direct-messaging ACT is Pronto-C2 [5]. Here, users derive a shared key from the advertised pseudonyms that is only identifiable to someone in possession of both pseudonyms. Since pseudonyms are rather long, they are uploaded to a bulletin board and only a link to the pseudonym is transmitted over BLE. If someone is infected, the shared key is published and distributed to all users. The authors did not consider the need for cover traffic, so while users do their risk scoring locally, this system leaks information. If no additional measures for hiding a users IP address are taken, the storage server will learn who interacted with one-another by monitoring who reads which pseudonyms from the bulletin board. This means the HA, the service operator and network operators might learn the social graph. They will not learn who is at risk. Users of the system will know which shared key belongs to which encounter and thereby be able to deanonymize infected users using their memory. The authors propose to use a blockchain for the server to ensure that no already published data can be deleted.

Similar to Pronto-C2 is the approach pursued by the Whisper Tracing Protocol [72]. Mobile devices scan for other compatible BLE devices and initiate a connection. For each connection, a session key is derived. These keys are published in case of an infection and a previous contact can query the central database. The matching system can be run both in a central and decentral manner, i. e., the matching can be done on the server or the end devices. This allows a trade-off between user privacy and the server being able to learn about the epidemiological spreading of the disease. The security risks for querying the system are the same as in ConTra Corona. Using decentral matching, the risks are the same as in Pronto-C2. Users will always learn which pseudonym infected them.

Deducing health status from traffic patterns is a big problem for systems relying on postboxes, 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 their ID.

5 COMMON ASPECTS

In this section, we discuss common aspects that can potentially arise in all BLE-based ACT systems. First security aspects of BLE-based proximity detection are discussed and how the rate of false negative to false positive can be regulated over transmission power. We then introduce various types of attacks and defense mechanisms, with a special focus on how to perform authentication and verification. Meta data can also leak information, therefore counter measures are evaluated from both a design as well as an operational security perspective. Last, features of ACT systems are discussed and a set of performance metrics for different architectures is provided.

5.1 Attacks against BLE

Here, problems with and attacks against BLE are discussed 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. This attack can not be mitigated [32].

5.1.2 Storage and Power Drainage Attacks. Another simple attack targets the exhaustion of battery power and storage of the end device by sending large amounts of BLE beacons [50]. This might make the ACT system unappealing to users, hindering widespread adoption. One solution used by BlueTrace [46] for filtering incoming broadcasts and finding new devices is to use a black list. GAEN [44] takes a sample of beacons at least every 5 minutes. The service responsible for handling received advertisements is specifically designed to be able to deal with large volumes, e.g. in public spaces. GAEN proposes using duplicate filters of the Bluetooth controller and hardware filters to deal with the problem.

5.1.3 Linking Advertisements. 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 [46, 110]. It has since been implemented by Google and Apple [43].

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. A simple solution proposed by West-Coast PACT [16] is to synchronize switching of pseudonyms between all users. Since this requires somewhat synchronized clocks in all end devices, it has instead been proposed to add jitter to the intervals between announcements [50]. This method is used by ConTra Corona [10].

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 [50] 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. No ACT system discussed in this survey takes measures against this variant of linking attacks.

5.1.4 Passive BLE-band 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 where used pseudonyms of infected individuals are published to allow local risk scoring. Most systems labeled as broadcast-based [16, 44, 53, 86, 93, 110] in this survey fall into this category. One mitigation measure is beacon secret sharing as proposed by the authors of DP-3T [110]. Here, instead of advertising the pseudonyms, only fragmented shares of pseudonyms are broadcast. The other side needs 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. ConTra Corona [10] improves this idea. The authors discovered that once the pseudonym that is shared changes, contacts with sufficient duration might not be recognized. They therefore make time slots of sequential pseudonyms overlap so that always two pseudonyms are advertised at the same time. Additional measures have to be taken so that the receiver knows which shares to combine to get a valid pseudonym.

Systems where communication is direct or that require the infected user to provide the pseudonyms of encounters are immune to this eavesdropping attack. Since a passive eavesdropper does not send out pseudonyms, they can not be contacted by the infected user. This is the case in the approaches of Cho et al. [18], CAUDHT [12], Pronto-C2 [5], Whisper [72], ConTra Corona [10] and TraceSecure [6]. The centralized designs of BlueTrace [46] and PePP-PT [57, 84] are also resistant to this attack.

5.1.5 Active BLE-band Eavesdroppers. An attacker might not be satisfied with passively collecting pseudonyms 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 a pseudonym originating from 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. ACT approaches where users do not learn which pseudonyms from their history of encounters belongs to an infected person, such BlueTrace [46] PePP-PT [57, 84], TraceSecure [6], the approach of Cho et al. [18] and ConTra Corona [10], are safe against this attack. Generally, all approaches discussed in this survey which were labeled as broadcast systems are vulnerable to this attack and also some message-based approaches.

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 False Positives and False Negatives

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

5.2.1 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 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 [46]. To ensure that an encounter was actually relevant, it is important that only those with a significant time span are taken into account. For example, PePP-PT NTK [84] has conducted field trials investigating different phone positions and distributions of people. DP3T focuses on detecting if distance is more or less than 2m instead of measuring the real distance [31].

5.2.2 False Negatives. Risky encounters might not be detected causing users at risk 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 [57].

5.3 Impersonation Attacks

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

5.3.1 One Contact Attack. Assume an attacker wants to find out if a person will later be diagnosed as positive. 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. The first method is used for example by Robert [57], the second one by BlueTrace [46] and ConTra Corona [10].

The authors of Robert [57] discuss a solution which is simple to implement for server-based ACT approaches. They propose probabilistic notifications, where for a small percentage of requests the server receives from clients for risk scoring, it always replies with a warning. This increases the false positive rate but provides plausible deniability. Users of Robert are also allowed to ever only receive a single positive answer, afterward the account will be deactivated. This makes it difficult for the attacker of launching multiple one contact attacks.

Another solution proposed by Gvili [50] applicable to all types of ACT 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. An observer will not be able to determine which transmissions come from which users, especially if the signal strength is varied. To the knowledge of the authors this has not been adapted by any ACT system discussed in the survey.

The authors of DP-3T [29] and ConTra Corona [10] have proposed remote attestation as mechanism to increase operational security. Operating systems allow backend servers to verify the integrity of devices and applications that want to communicate with them by using Google SafetyNet or iOS DeviceCheck. These mechanisms allow the identification of altered apps, which are needed for the execution of a one contact attack if there are other user around.

Epione [108] suggests rate limiting the the number queries that can be send by a user in ACT systems that rely on queries to the server. They also discuss uploading a daily cryptographic hash over the encounter history using a Merkle tree to ensure that not only a subset of encounters is used in a query.

5.3.2 Replay and Relay Attacks. One way for an attacker to make random people believe they are infected is by recording pseudonyms and replaying BLE messages. Messages can for example be collected at high risk areas like a testing center and be emitted at a different location frequented by a certain target or demographic. Using a dedicated antenna, the attacker can receive advertisements within 20-100m range [110]. To limit the impact of replay attacks most approaches [16, 57, 86, 109] encode the epoch of the encounter in the transmitted pseudonym. Centralized systems like BlueTrace [46] and PePP-PT [57, 84] can check when an encounter was recorded and whether the recorded pseudonym was actually in use at that time. It has been argued that replaying a single pseudonym might not be sufficient to surpass the threshold duration and be counted as close contact. Broadcast systems allow users to check themselves if they recorded a corresponding encounter for this time slot. The unlinkable design of DP-3T also cryptographically links pseudonyms with their epoch [110]. Some direct messaging approaches, e.g. CAUDHT [12], Pronto-C2 [5], include the sender's pseudonym at the epoch of the encounter or a distinct shared key 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 relays the collected pseudonyms during the same epoch in which they were collected. As mitigation Vaudenay [111] proposes to switch from passively exchanging pseudonyms through broadcasts to an active protocol. This is done by the Whisper protocol [72]. It has been warned that an active exchange of messages is less secure than one-way communication where users simply send advertisements and listen to other advertisements as it opens the door for new types of attacks against the end device [5]. Energy consumption also increases in such a scenario. Some works [50, 85] propose to use coarse (GPS) location data in the broadcast of the pseudonyms. This allows the receiver to figure out if the sender is actually close. A similar solution is also used by Hashomer [86]. They introduce a message authentication code to prove the authenticity of the geo-location encoded in the BLE message. The BLE message can also indicate that no location information is available by using a specific pseudonym. If the majority of users do send location information over BLE, relay attacks are mostly mitigated.

5.4 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.

5.4.1 Authenticating Uploads. Ensuring that a user is indeed infected while enforcing privacy is an important aspect. If there is no control over who is capable to upload data into the system, trolling and planting fake data becomes possible. This makes warnings issued by the ACT system unreliable. The simplest solution to this problem, taken for example by Epione [108] and ConTra Corona [10], has health care providers collecting data that is then uploaded to the server. This way the system can be sure that as long as the health care providers are trusted, the uploaded data is authentic. Most ACT approaches [5, 12, 44, 46, 57, 84, 93, 108, 110] use token systems that allow infected users to upload their data to the server after having received confirmation from a doctor. GAEN relies on doctors to provide their diagnosed patients with a verification code to enable uploads [45]. The authors of DP-3T discuss multiple types of token systems for upload verification [33]. Tokens can either be handed out when the infection is verified or if an additional activation mechanism is used, at the time of testing. Another option mentioned by the authors is that users commit data when being tested which will be uploaded later. It they test positive, the HA authorises the upload. This mechanism ensures that the data is not tempered with after the infection is verified. This also stops users from giving their upload token to others.

Some direct messaging approaches like CAUDHT [12] and Pronto-C2 [5] use blind signatures to prove to users that data was sent by infected individuals. This way the health authority does not learn the content to be signed, but users can fetch valid signatures and recipients can be certain about the authenticity of a warning. In this situation, a hacked malicious server (without access to the HA's private key) can only delete messages but not insert new ones. To know who to issue blind signature to, again a token based access mechanism is used.

CONTAIN [53] solves the issue of authenticating uploads by having infected users generate a new public key pair that will be singed by the HA to create an infection certificate. This certificate can be provided on upload to verify the infection. To ensure privacy, this mechanisms relies on non-collusion of HA and server.

5.4.2 Verifying Encounters. Imagine a black market where people offer money for faking encounters of a target Tiffany 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 Tiffany. The attack can be stopped by having the client check if they have recorded a corresponding encounter. This is done by design by broadcast-based ACT systems where the advertised pseudonyms of infected individuals are published, so all the broadcast systems mentioned in this paper [16, 44, 53, 86, 93, 110]. CAUDHT [12], Pronto-C2 [5] and Whisper [72] have similar sanity checks. Server-based approaches, like BlueTrace [46] and PePP-PT [57, 84] do not defend against this attack since risk scoring is done by the server. A malicious user can upload pseudonyms of the target Tiffany. The server will recognize the pseudonyms and send a notification to Tiffany. She will not be able to tell the difference between a misbehaving server, fake data inserted by other users and a real warning. The same goes for ConTra Corona [10], except that the server only publishes the user's warning identity instead of sending a message.

Liu et al. [69] propose their own solution to the problem of verifying if a close contact actually occurred. 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. Many designs do not want to use active protocols as this can make it easier for an attacker to exploit the device (see Section 2.3) [5]. This approach has therefore not been employed by any of the ACT systems discussed in the survey.

The authors of Epione [108] use cryptographic hashes to tackle the problem of verifying encounters. Users compute a daily hash for their history of contact which they upload to the server. To verify the hash, a zero knowledge proof has to be used. This mechanism ensures that an attacker can not forge queries or only use a subset of the data.

5.4.3 Incomplete Reports. Users want to have control over what they report, so that no sensitive data is leaked. They can always turn off BLE when they do not want any data to be collected. Whether or not an infected user provides their data for ACT is mostly voluntary. Some, like East-Cost PACT [93], Hashomer [86] and Pronto-C2 [5], additionally consider the option for users to only upload some data to the server. This leaves room for extortion, as infected people could blackmail other users so they are not included in the upload.

5.5 Metadata

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

5.5.1 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. In decentralized systems users might not want to reveal the fact that they participate. Also, depending on the actual authentication mechanisms, users might want to ensure that uploaded data (like past pseudonyms) is not linkable to their identity. For this purpose anonymisation networks like Tor [27] or mix networks [22] can be used. This is employed by Cho et al [18], CAUDHT [12], Pronto-C2 [5], ConTra Corona [10] and CONTAIN [53]. 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 [80], mix networks do not have this drawback, but are slower at delivering messages.

Another option to mitigate IP Address leakage is used by message-based TraceSecure [6]. Here, a messaging service receives encrypted messages by the government and lets users download the ones that are designate for them. To ensure privacy, the message service is not allowed to collude with the service which places messages there.

Epione [108] takes a different approach to the problem. The assumption is that the health care provider (for example the facility where the user got tested) knows the infected user's identity. The infected user can therefore freely communicate with the health care provider and upload their encrypted data there. The health care provider will collect data from multiple infected users and shuffles before uploading everything to the server responsible for ACT. The health care provider works as anonymisation proxy and is therefore not allowed to collude with the ACT server.

5.5.2 Traffic Analysis. Anonymisation networks do not only hide IP addresses but also stop an attacker who observes the network from finding out who is infected. Cho et al. [18], CAUDHT [12], Pronto-C2 [5], ConTra Corona [10] and CONTAIN [53] use Tor or mix networks to avoid traffic analysis. But anonymisation networks are known to have performance and scaling issues [33]. The authors of PePP-PT NTK argue, that the current Tor network is not equipped to support the expected user basis of an ACT system [84]. Therefore mechanisms are used which leak the users IP address, but still defend against network observers. To ensure a network observer does not learn if an upload contains real data which indicates that the sender is infected, DP-3T regularly uploads dummy data [29]. If a user is diagnosed and uploaded their real data, it is necessary that the app continues downloading keys and making fake requests. Message-based TraceSecure [6], Cho et al. [18] and CAUDHT [12] also use cover traffic to hide which messages are real. HE-based Tracesecure and other cryptographic approaches ensure that both, a warning and an empty message, look the same for everyone except the designated user.

5.5.3 Leakage through Upload Timing. Another type of metadata that might be used to derive information is timing. When uploading data that should not be linked by the server, it is necessary to also induce jitter. This is for example done in Robert [57] to break the link between two uploads from the same infected user. The authors also consider mix networks and additional servers with secure hardware modules for this purpose. Additionally to jitter, mix networks and onion-routing, Pronto-C2 [5] suggests that cover traffic is helpful to defend against this type of linking attacks. It becomes unclear for the attacker which data is real. This method can only be applied to systems where the server can not tell real data from generated date. CAUDHT [12] and the approach of Cho et al. [18] also fall into this category. Epione [108] solves this issue by having the health care provider collect data from multiple infected users and shuffles them before uploading. This stops the ACT server from knowing which data points belong to which user.

5.6 Hacking, Backdoors and Malware

ACT systems generally rely on apps being installed on the user's smartphone. Like in any kind of IT environment, both underlying hardware and software can be vulnerable. Therefore regular updates are mandatory to ensure privacy. To guarantee that no other installed applications can spy on the ACT app, it has been suggested that employing Trusted Platform Modules (TPM) would help [111]. Remote attestation mechanisms available in most smartphones are also useful to detect hacked devices [29].

But hackers can also attack servers directly and use log files to identify infected users by the IP address of their upload. To prevent this kind of privacy leak, East-Coast PACT [93] suggests not maintaining logs that might leak the identity of infected users. The authors also suggest that relying on anonymisation networks does hide information in log file which might be of interest for the hackers. This approach is also used by CAUDHT [12], CONTAIN [53], ConTra Corona [10] and Pronto-C2 [5]. DP-3T relies on the fact that dummy traffic used to defend against network observers will also hide real uploads in log files [29].

Users' trust is an important building block of ACT systems. It has often been argued that making code open source is a requirement to ensure that an ACT system is trustworthy [10, 18, 53]. Having code freely accessible allows independent security researchers to check that no backdoors have been implemented and that the app is not actually malware. Open Source code is available for BlueTrace [46], PePP-PT NTK [84], Robert [57], DP-3T [109], Hashomer [86] and Covid-Watch [19]. 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. The authors of PePP-PT NTK [84] discuss the usage of trusted execution environment on the server side to verify to users that the source code running is the same as the one that is openly available.

5.7 Proving Risk

It has been suggested that users who have received a warning should have a right to be tested. This is especially of interest in places where testing capacities are sparse. For centralized systems such as BlueTrace [46] and PePP-PT [57, 84] it is easy to determine who is eligible for a test, as servers provide some degree of validation. For systems where the server is not informed about results from risk assessment, the process is more complicated. Even if a user receives a notification, they have to prove they are not simply forging encounters and notifications to get tested. For systems that rely on asymmetric key cryptography, the possession of a private key corresponding to an at-risk public key can be used as proof. This applies for CAUDHT [12], the proposal of Cho et al. [18], Pronto-C2 [5] and message-based TraceSecure [6].

To prove exposure, Hashomer [86], 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 if the user becomes infected. Users that want to prove they are at risk can present the corresponding collected pseudonym. Using the verification key, the HA can figure out if the collected pseudonym 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 to verify ownership. To discourage people from giving away their proof, u can include a timestamp and the user's real identity.

The authors of DP-3T [110] mention that the use of cryptographic primitives makes it possible to include a feature for proving one's risk. But since their main focus lies on retaining interoperability with the GAEN API by Google and Apple, they only discuss mechanisms that rely on the integrity of the ACT application and are therefore easily circumvented by a tech-savvy user.

5.8 Dealing with International Travel

To facilitate cooperation between different states, PePP-PT includes a system for federation between different health authorities [57, 84]. A country code is added to the pseudonym when it is transmitted. BlueTrace also supports federation, in a similar manner [46]. DP-3T [30] pays particular attention to the aspect of interoperability across boarders, allowing users to enter regions they will travel to or have returned from. When diagnosed as infected, users upload their history. If they indicated a travel, the backend will communicate the pseudonyms of the user to the backend responsible for that specific region. The mechanism requires backends to trust one another to function properly. GAEN [87] is generally capable of providing tracing internationally for all region specific apps that build on its API.

5.9 Performance Considerations

There are multiple aspects that are relevant when considering performance. How often pseudonyms are switched is one issue common to all BLE-based ACT systems. To mitigate tracking, pseudonym advertising epochs are generally less than 20 minutes. The authors of DP-3T [110] come to the conclusion that their various approaches require between 4.8 MB to 6.9 MB storage for received pseudonyms when assuming that received data has to be stored for 14 days and an estimate of 140k different observations are made in that time. They consider an epoch of 15 minutes. The same value is also used by BlueTrace [46] and Robert [57]. A small value for the rotation period improves security, as it makes tracking harder. At the same time it becomes more difficult to recognize relevant encounters, because successive pseudonyms can not be linked. But 15 minutes is the minimal duration that needs to pass between two pseudonyms as it is also the rotation period of random addresses in Bluetooth (as of version v5.1). This means choosing a smaller value does not improve privacy but increases the amount of data to be communicated [57].

Performance varies greatly between different groups of contact tracing systems. Regarding the cryptographic approaches, we will only talk about Epione [108] as it has the best performance of those discussed in this survey. The used setup were two servers (Intel(R) Xeon(R) E5-2699 v3 2.30GHz CPU and 256 GB RAM) and a cache. The authors did not use a smartphone as client device but a computer with the same specifications as the server. The evaluations are built on the assumption that the server has 10^6 pseudonyms each day, which corresponds to 5000 new cases a day. Users get a new pseudonym every 15 min and query once per day. The authors were able to achieve a client-side runtime of 121 ms and a server-side runtime of 1635 ms. They required 37 MiB of network traffic per user. This indicates that cryptographic approaches are feasible but not scalable to nation-wide deployment.

Little performance measurements have been done for systems using messaging based approaches. The authors of TraceSecure [6] have calculated that their approach requires several hundred gigabyte of storage on the server side for the government. The other server can be smaller.

BlueTrace [46], as a system that relies on server-based risk assessment, has only a small overhead. The authors found out that creating pseudonyms on the device greatly increases the computational requirements. They therefore decided to compute them online and have users fetch new pseudonyms regularly. To deal with client side pseudonym generation, Hashomer uses symmetric key instead of public key cryptography. Expensive functions that require costly HMAC operations are run less often.

Performance issues of broadcast-based systems stem from the large amount of data that has to be downloaded daily. The authors of Hashomer calculated, that 5.4 MBytes of data have to be downloaded by users daily if 1000 people are newly infected per day. To reduce this, they suggest instead of sending keys for each epoch, keys for each day can also be used. Their system supports both types. In the case of daily keys, the amount of data is reduced to 224 KBytes. The authors of DP-3T [110] come to the conclusion the serving the daily download requests are easily manageable using a CDN. For 1000 new cases they are far below 1MB of communicated data for all of their designs.

Running an application in the foreground can be very costly. As we have seen earlier in section 2.3, to run BLE scanning in the background special permissions and changes to the operating system are needed. Otherwise the battery is drained or in the case of iOS devices, scanning does not work at all. Google and Apple are the two main providers of operating systems for smartphones worldwide [73]. Since they have implemented their own API following a broadcast-based design, any other application not using this API effectively drains the battery.

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 homomorphic encryption 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 mitigations. 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)": For known pseudonyms. "(2)": Cryptographic overhead on end devices. "(3)": Cryptographic and polling overhead on end devices.

		Name	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 [46]	Trusted	x	x			x	
		PePP-PT (NTK [84], Robert [57])	Trusted	x	x			x	
		Aarogya Setu [77]	Trusted	x	x			x	GPS+BLE
	Cryptographic	EPIC [3]	Semi-honest				x		Passively collected Wifi+Bluetooth advertisements HE
		HE-based TraceSecure [6]	Semi-honest				x		HE
		Berke et al. [7]	Semi-honest			(1)	x		GPS, MPC (PSI)
		Reichert et al. [91]	Semi-honest/ malicious			(1)	x		GPS, MPC
		Demirag et al. [24]	Semi-honest				x		MPC (PSI-CA)
		Epione [108]	Semi-honest/ malicious				x		HE+MPC (PSI-CA)
Client-side	Broadcast	DP-3T [109]	Semi-honest			(1)	(2)		
		Apple+Google(GA-ACT) [44]	Semi-honest			(1)	(2)		
		CONTAIN [53]	Trusted			(1)	Proto 1: (2)		
		East-Coast PACT (Rivest et al.) [93]	Semi-honest			(1)	(2)		
		West-Coast PACT (Chan et al.) [16]	Semi-honest			(1)	(2)		
		Covid-Watch [19]	Semi-honest			(1)	(2)		
		Hashomer [86]	Semi-honest			(1)	(2)		
	Direct messaging	Cho et al. [18]	Semi-honest			Gives time period of encounter	(3)	Uses cover traffic	
		CAUDHT [12]	Semi-honest			Gives the pseudonym of infected person used in encounter	(3)	Uses cover traffic	DHT instead of central server
		Pronto-C2 [5]	Semi-honest			Gives the pseudonym of infected person used in encounter		x	Can be implemented with blockchain
		TraceSecure (messaging approach) [6]	Semi-honest	x	partially	Gives time period of encounter		Uses cover traffic	Relies on HA and government not to cooperate.
		ConTra Corona [10]	Trusted		x	Gives time period of encounter			Relies on HA and matching server not to cooperate.
		Whisper (central) [72]	Trusted		anonymous				Uses active connection protocol.
		Whisper (decentral) [72]	Semi-honest			Gives the pseudonym of infected person used in encounter			Uses active connection protocol.

REFERENCES

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

[2] 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. March 2020.

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

[4] 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.

[5] 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.

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

[7] 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).

[8] Berliner Zeitung. 2020. Raus aus dem Lockdown - Corona-Warn-App steht zum Download bereit, aber es gibt noch Forderungen. www.berliner-zeitung.de/zukunft-technologie/corona-warn-app-starttermin-am-dienstag-steht-aber-es-gibt-noch-forderungen-li.87669 Accessed: 17. September 2020.

[9] Stefano Bertuletti, Andrea Cereatti, Ugo Della Croce, Michele Caldara, and Michael Galizzi. 2016. Indoor distance estimated from Bluetooth Low Energy signal strength: Comparison of regression models. In *IEEE Sensors Applications Symposium*. IEEE, Catania, Italy, 1–5. https://doi.org/10.1109/SAS.2016.7479899

[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/ Accessed: 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] Fabian Buder et al. 2020. Adoption Rates for Contact Tracing App Configurations in Germany. www.nim.org/en/research/research-reports/adoption-rates-contact-tracing-app Accessed: 8. September 2020.

[14] 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.

[15] 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. April 2020.

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

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

[18] 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).

[19] Covid Watch. 2020. Covid Watch. www.covid-watch.org Accessed: 07. April 2020.

[20] 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.

[21] 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, Darmstadt, Germany, 218–231.

[22] George Danezis, Roger Dingledine, and Nick Mathewson. 2003. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *IEEE Security and Privacy*. IEEE Computer Society, Berkeley, CA, USA, 2–15.

[23] Helen Davidson. 2020. China's coronavirus health code apps raise concerns over privacy. The Guardian. www.theguardian.com/world/2020/apr/01/chinas-coronavirus-health-code-apps-raise-concerns-over-privacy Accessed: 08. September 2020.

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

[25] 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 Accessed: 30. April 2020.

[26] 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. March 2020.

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

[28] 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.

[29] DP-3T. 2020. Best Practices Operational Security for Proximity Tracing. github.com/DP-3T/documents/blob/master/DP3T-BestPracticesforOperationSecurityinProximityTracing.pdf Accessed: 09. September 2020.

[30] DP-3T. 2020. Decentralized Proximity Tracing Interoperability Specification. github.com/DP-3T/documents/raw/master/DP3T-InteroperabilityDecentralizedProximityTracingSpecification(Preview).pdf Accessed: 09. September 2020.

[31] DP-3T. 2020. DP-3T Exposure Score Calculation - Summary. github.com/DP-3T/documents/raw/master/DP3T-ExposureScoreCalculation.pdf Accessed: 09. September 2020.

[32] DP-3T. 2020. Privacy and Security Attacks on Digital Proximity Tracing Systems. github.com/DP-3T/documents/Securityanalysis/PrivacyandSecurityAttacksonDigitalProximityTracingSystems.pdf Accessed: 11. September 2020.

[33] DP-3T. 2020. Secure Upload Authorisation for Digital Proximity Tracing. github.com/DP-3T/documents/blob/master/DP3T-UploadAuthorisationAnalysisandGuidelines.pdf Accessed: 09. September 2020.

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

[35] 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. www.osf.io/preprints/lawarxiv/yc6xu. Accessed:16. May 2020.

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

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

[38] Luca Ferretti et al. 2020. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science* 368, 6491 (2020).

[39] Simon M Firestone, Robert M Christley, Michael P Ward, and Navneet K Dhand. 2012. Adding the spatial dimension to the social network analysis of an epidemic: Investigation of the 2007 outbreak of equine influenza in Australia. *Preventive veterinary medicine* 106, 2 (2012), 123–135.

[40] 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.

[41] 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.

[42] 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, New York, New York, USA, 218–229.

[43] 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 Accessed: 18. May 2020.

[44] Google and Apple. 2020. Privacy-Preserving Contact Tracing. www.apple.com/covid19/contacttracing Accessed: 11. September 2020.

[45] Google Inc. 2020. Exposure Notifications verification server. https://developers.google.com/android/exposure-notifications/verification-system Accessed: 09. September 2020.

[46] Government of Singapore. 2020. BlueTrace. www.bluetrace.io Accessed: 26. April 2020.

[47] Government of Singapore. 2020. TraceTogether. www.tracetogether.gov.sg Accessed: 06. April 2020.

[48] 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 Accessed: 26. April 2020.

[49] Rajan Gupta, Manan Bedi, Prashi Goyal, Srishti Wadhera, and Vaishnavi Verma. 2020. Analysis of COVID-19 Tracking Tool in India: Case Study of Aarogya Setu Mobile Application. *Digital Government: Research and Practice* 1, 4 (2020), 1–8.

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

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

[52] 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. March 2020.

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

[54] Robert Hinch et al. 2020. Effective configurations of a digital contact tracing app: A report to NHSX. github.com/BDI-pathogens/covid-19_instant_tracing/blob/master/Report Accessed: 8. September 2020.

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

[56] INRIA. 2020. The StopCovid project, a digital solution to contribute to the citizens' fight against the Covid19 epidemic. www.inria.fr/en/le_projet_stopcovid Accessed: 10. September 2020.

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

[58] Andrea Jelinek. 2020. EDPB Letter concerning the European Commission's draft Guidance on apps supporting the fight against the COVID-19 pandemic. edpb.europa.eu/our-work-tools/our-documents/letters/edpb-letter-concerning-european-commissions-draft-guidance-apps_en Accessed: 17. September 2020.

[59] 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.

[60] Malek Karaim, Mohamed Elsheikh, and Aboelmagd Noureldin. 2018. *GNSS Error Sources*. IntechOpen, London, UK, Chapter 2. https://doi.org/10.5772/intechopen.75493

[61] 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.

[62] Michael Klenk and Hein Duijf. 2020. Ethics of Digital Contact Tracing and COVID-19: Who Is (Not) Free to Go? *Ethics and Information Technology* (2020).

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

[64] Mirjam E Kretzschmar, Ganna Rozhnova, Martin CJ Bootsma, Michiel van Boven, Janneke HHM van de Wijgert, and Marc JM Bonten. 2020. Impact of delays on effectiveness of contact tracing strategies for COVID-19: a modelling study. *The Lancet Public Health* 5, 8 (2020), e452–e459.

[65] Adam J Kucharski et al. 2020. Effectiveness of isolation, testing, contact tracing and physical distancing on reducing transmission of SARS-CoV-2 in different settings. *medRxiv* (2020).

[66] Seungho Kuk, Junha Kim, Yongtae Park, and Hyogon Kim. 2018. Empirical Determination of Efficient Sensing Frequencies for Magnetometer-Based Continuous Human Contact Monitoring. *Sensors* 18, 5 (2018), 1358. https://doi.org/10.3390/s18051358

[67] 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. April 2020.

[68] 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.

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

[70] 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.

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

[72] Lucien Loiseau et al. 2020. Whisper Tracing Version 3 - an open and privacy first protocol for contact tracing. docsend.com/view/nis3dac Accessed: 10. September 2020.

[73] Macworld. 2019. iPhone vs Android market share. www.macworld.co.uk/feature/iphone/iphone-vs-android-market-share-3691861/ Accessed: 16. September 2020.

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

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

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

[77] National Informatics Centre, Ministry of Electronics & Information Technology, Government of India. 2020. Aarogya Setu Mobile App. www.mygov.in/aarogya-setu-app/ Accessed: 16. September 2020.

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

[79] 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, Sapporo, Japan, 1–8.

[80] 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, San Diego, CA, USA.

[81] Andrea Nuzzo, Can Ozan Tan, Ramesh Raskar, Daniel C. DeSimone, Suraj Kapa, and Rajiv Gupta. 2020. Universal Shelter-in-Place Versus Advanced Automated Contact Tracing and Targeted Isolation: A Case for 21st-Century Technologies for SARS-CoV-2 and Future Pandemics. *Mayo Clinic Proceedings* 95, 9 (2020), 1898 – 1905. https://doi.org/10.1016/j.mayocp.2020.06.027

[82] Patrick Howell O'Neill. 2020. No, coronavirus apps don't need 60% adoption to be effective. MIT Technology Review. www.technologyreview.com/2020/06/05/1002775/covid-apps-effective-at-less-than-60-percent-download/ Accessed: 08. September 2020.

[83] PePP-PT e.V. i.Gr. 2020. PePP-PT. www.pepp-pt.org Accessed: 05. April 2020.

[84] PePP-PT e.V. i.Gr. 2020. PePP-PT Documentation. www.github.com/pepp-pt/pepp-pt-documentation Accessed: 26. April 2020.

[85] Krzysztof Pietrzak. 2020. Delayed Authentication: Preventing Replay and Relay Attacks in Private Contact Tracing. Cryptology ePrint Archive, Report 2020/418.

[86] Benny Pinkas and Eyal Ronen. 2020. Hashomer Crypto Reference. github.com/eyalr0/HashomerCryptoRef. Accessed: 12. May 2020.

[87] Presse- und Informationsamt der Bundesregierung. 2020. Corona warn app - Frequently asked questions. www.bundesregierung.de/breg-de/themen/corona-warn-app/corona-warn-app-englisch/corona-warn-app-faq-1758636. Accessed: 14. September 2020.

[88] Mimonah Al Qathrady, Ahmed Helmy, and Khalid Almuzaini. 2016. Infection tracing in smart hospitals. In *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, New York, USA, 1–8.

[89] Aswin N. Raghavan, Harini Ananthapadmanaban, Manimaran Sivasamy Sivamurugan, and Balaraman Ravindran. 2010. Accurate mobile robot localization in indoor environments using bluetooth. In *ICRA*. IEEE, Anchorage, Alaska, USA, 4391–4396.

[90] Ramesh Raskar et al. 2020. Comparing manual contact tracing and digital contact advice. arXiv:2008.07325 [cs.CY]

[91] 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.

[92] Mohamed Er Rida, Fuqiang Liu, Yassine Jadi, Amgad Ali Abdullah Algawhari, and Ahmed Askourih. 2015. Indoor Location Position Based on Bluetooth Signal Strength. In *ICISCE*. IEEE, Shanghai, China, 769–773.

[93] Ronald L. Rivest et al. 2020. The PACT protocol specification. pact.mit.edu/. Accessed: 13. May 2020.

[94] Miguel Rodriguez, Juan P Pece, and Carlos J Escudero. 2005. In-building location using bluetooth. In *IWWAN*. Springer, Nice, France.

[95] Jilian A Sacks et al. 2015. Introduction of mobile health tools to support Ebola surveillance and contact tracing in Guinea. *Global Health: Science and Practice* 3, 4 (2015), 646–659.

[96] Sanjay Sareen, Sandeep K. Sood, and Sunil Kumar Gupta. 2018. IoT-based cloud framework to control Ebola virus outbreak. *J. Ambient Intell. Humaniz. Comput.* 9, 3 (2018), 459–476. https://doi.org/10.1007/s12652-016-0427-7

[97] James Scott, Pan Hui, Jon Crowcroft, and Christophe Diot. 2006. Haggle: A Networking Architecture Designed Around Mobile Users. In *WONS*. IFIP, Les Ménuires, France.

[98] 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.

[99] 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. Accessed: 26. March 2020.

[100] 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. March 2020.

[101] Nigel P. Smart. 2016. *Cryptography Made Simple*. Springer, Switzerland.

[102] Chanjuan Sun and Zhiqiang Zhai. 2020. The efficacy of social distance and ventilation effectiveness in preventing COVID-19 transmission. *Sustainable Cities and Society* 62 (2020), 102390. https://doi.org/10.1016/j.scs.2020.102390

[103] The Directorate of Health and The Department of Civil Protection and Emergency Management. 2020. Join the tracing team! Contagion tracing is a community affair. www.covid.is/app/en Accessed: 10. September 2020.

[104] The MITRE Corporation. 2020. CVE-2020-0022. www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-0022. Accessed: 02. June 2020.

[105] 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 Accessed: 04. May 2020.

[106] 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 Accessed: 06. April 2020.

[107] Simon Trang, Manuel Trenz, Welf H. Weiger, Monideepa Tarafdar, and Christy M.K. Cheung. 2020. One app to trace them all? Examining app specifications for mass acceptance of contact-tracing apps. *European Journal of Information Systems* 0, 0 (2020), 1–14. https://doi.org/10.1080/0960085X.2020.1784046 arXiv:doi.org/10.1080/0960085X.2020.1784046

[108] Ni Trieu, Kareem Shehata, Prateek Saxena, Reza Shokri, and Dawn Song. 2020. Epione: Lightweight Contact Tracing with Strong Privacy. *CoRR* abs/2004.13293 (2020).

[109] Carmela Troncoso et al. 2020. Decentralized Privacy-Preserving Proximity Tracing. www.github.com/DP-3T/documents Accessed: 28. May 2020.

[110] 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 Accessed: 28. May 2020.

[111] Serge Vaudenay. 2020. Analysis of DP3T. Cryptology ePrint Archive, Report 2020/399.

[112] Serge Vaudenay. 2020. Centralized or Decentralized? The Contact Tracing Dilemma. Cryptology ePrint Archive, Report 2020/531.

[113] 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).

[114] Eiko Yoneki. 2011. FluPhone study: virtual disease spread using haggle. In *CHANTS@MobiCom*. ACM, Las Vegas, NA, USA, 65–66.

[115] Kuan Zhang, Xiaohui Liang, Jianbing Ni, Kan Yang, and Xuemin Sherman Shen. 2018. Exploiting Social Network to Enhance Human-to-Human Infection Analysis without Privacy Leakage. *IEEE Transactions on Dependable and Secure Computing* 15, 4 (2018), 607–620.

[116] Zhaoyang Zhang, Honggang Wang, Xiaodong Lin, Hua Fang, and Dong Xuan. 2013. Effective epidemic control and source tracing through mobile social sensing over WBANs. In *2013 Proceedings IEEE INFOCOM*. IEEE, Turin, Italy, 300–304.

[117] Sheng Zhou and John K. Pollard. 2006. Position measurement using Bluetooth. *IEEE T CONSUM ELECTR* 52, 2 (2006), 555–558.