Decentralized Contact Tracing Using a DHT and Blind Signatures

Samuel Brack*, Leonie Reichert*, Björn Scheuermann*†
*Humboldt University of Berlin, Department of Computer Science
{samuel.brack,leonie.reichert,scheuermann}@informatik.hu-berlin.de
†Alexander von Humboldt Institute for Internet and Society, Berlin
bjoern.scheuermann@hiig.de

***Abstract*—Contact tracing is a promising approach to combat the COVID-19 pandemic. Various systems have been proposed to automatise the process. However, user privacy was not a major design goal in most of these systems. Other designs rely heavily on a centralised server or reveal significant amounts of private data to health authorities. We propose CAUDHT, a decentralized peer-to-peer system for contact tracing. The central health authority can focus on providing and operating tests for the disease while contact tracing is done by the system's users themselves. We use a distributed hash table to build a decentral messaging system for infected patients and their contacts. With blind signatures, we ensure that messages about infections are authentic and unchanged. A strong privacy focus enables data integrity, confidentiality, and privacy.**

***Index Terms*—COVID-19, Contact Tracing, Privacy-enhancing technologies, DHT, Blind Signatures**

I. INTRODUCTION

The current COVID-19 pandemic shows that our modern globalized world can be heavily affected by a quickly spreading, highly infectious, deadly virus in a matter of weeks. It became apparent that manual contact tracing and quarantining of suspects can only be effective in the first days of the spread before the exponential growth overwhelms the health authorities. Shutdowns of entire countries thus are a popular and drastic method to slow down infection rates in order to not overwhelm emergency capacities. While such shutdowns are effective, they also severely impact social and economical routines in the affected areas.

By automating tracing processes and quarantining everyone who came in contact with infected people, as well as arriving travelers, it should be possible to quickly loosen lockdown measures. Bluetooth tracing has emerges as the most suitable method for tracking infections of airborne diseases such as COVID-19. Singapore was first to implement such a scheme, allowing the government to identify possible infections and forcing people into quarantine [1]. Due to privacy concerns and data protection laws European and American initiatives aim to built somewhat more privacy-preserving systems. But the current plans still leave room for improvement, as they mostly rely on central servers and broadcasts of device-dependent identifiers. There have been calls for decentralization and demands regarding properties privacy-preserving contact tracing systems should fulfill [2].

To approach this we propose CAUDHT (Contact tracing Application Using a Distributed Hash Tables), a system for Distributed Contact Tracing using privacy-preserving messaging to enable notification. Our main contributions are:

- An Identification and formulation of privacy risks of contact tracing and
- the design and analysis of a decentralized privacy-preserving approach to contact tracing.

To build an efficient and scalable decentralized contact tracing system we use a distributed hash table (DHT) operated by all users. The DHT allows us to implement a messaging service for encrypted and signed messages between users to inform each other about infection statuses. Additionally, infected patients are able to prove their infection status without revealing their location history by requesting a *blind signature* from the health authority. This measure ensures that users can trust an infection warning is not generated by a malicious party trying to spread misinformation.

II. RELATED WORK

Contact Tracing is the process of identifying potentially infected people by analyzing a patients history of social contacts. This has been done for epidemics such as HIV [3] or Ebola [4]. Stochastic analysis and real world experience have proven its usefulness [3]–[5]. For the 2020 COVID-19 pandemic, health authorities (HA) in Germany were able to hold off the disease for a few weeks by manually tracing contacts and quickly quarantine infected people [6]. For contact tracing to be effective, the number of identified cases has to grow faster than the number of new infections [3]. With growing numbers of patients this process requires to be automated to stop the spreading.

Systems using Bluetooth communication for automatic contact tracing has been first proposed by Altuwaiyan et al. in 2018 [7]. Users of their system passively collect information about surrounding Bluetooth and WIFI IDs by doing scans. Scans of infected individuals are uploaded to a central server. Pairwise matching scores between user data and the database are regularly calculated to identify contacts. Homomorphic encryption is used to allow users to query the database in a private manner. While the system does not leak information about the results to the HA, it reveals timestamps of scans and requires infected people to reveal their data to the HA.

Various systems for contact tracing have been implemented and rolled out in the past few months. The app called *TraceTogether* [8], [9] released by the government of Singapore, which has been the first running official system for automatic contact tracing, uses some ideas of Altuwaiyan et al. Here, users broadcast time-dependent IDs using Bluetooth while continuously scanning their surroundings. The scan history is stored locally, while own used IDs are uploaded to a server. When a person falls ill, the HA can ask or force for the history to be uploaded to their servers. It can then determine who has been in contact with this individual by searching for IDs from the history in their database. The corresponding users can then be informed about their possible infection status and the HA can prescribe testing as well as quarantine.

Other states have followed the example of Singapore by publishing plans for similar applications. The pan-European initiative *PEPP-PT* [10] aims at improving the system by broadcasting IDs from the history of infected people. While this ensures that users can be certain that their infection status is not revealed to the HA, broadcasts are expensive and users can use the information to deanonymize patients. Troncoso et al. have proposed in an whitepaper [11] to use hash chains of IDs to ensure pseudo-randomness to improve privacy for Bluetooth based contact tracing systems relying on central servers. Initiatives for creating more privacy preserving contact tracing exist among others in the United States [12]–[14], India [15] and Austria [16].

Cho et al. proposed to use private messaging for notification of possible contacts after collecting Bluetooth IDs [9]. Messages containing the infection status are sent to a private mailbox located on a central server via a proxy servers. Users regularly query all mailboxes corresponding to their past IDs for new messages. Messages also have to be sent even when the sender is not infected. The authors do not discuss scalability issues of their idea.

Instead of using Bluetooth other sensors could also be used. The Israeli state for example uses geolocation information for tracking [17]. More privacy-preserving variants have been proposed since [18], [19]. It is also possible to use magnetometer reading in smartphones to correlate similar locations without revealing the actual coordinates where they occurred [20].

III. Attacker Model

To understand the security requirements, we discuss several threats to a naive automatic contact tracing system.

A. Health Authority

The HA is assumed to offer the medical tests if a person is infected or not. It can try to deanonymize users' IDs to trace possible new infections without respecting users' privacy. It can also try to link the IDs submitted by a single infected person with those of other people to identify common contacts (and thus location history).

B. Infected Users

Infected users can try to spread panic and report random IDs as past contacts, even though they have not met. This could force the corresponding users into quarantine or harm their reputation.

C. Uninfected Users

Healthy users can try to report themselves as infected to force their past contacts into quarantine. Users can also try to identify infected patients' identities by comparing published IDs with IDs they encountered in the past.

IV. System Design

Contact tracing system often leverage the fact that a country's central HA is involved in determining if a person contracted a dangerous disease or not. However, this results in a loss of privacy as it requires the contact history of infected people to be revealed to the HA in order to trace and inform possible disease carriers. Systems using ephemeral Bluetooth IDs that change every few hours for identifications such as TraceTogether [8] also leak information. Here, a malicious HA (or an attacker gaining access to the HA's collected data) would be capable of deriving some information from the transmitted contacts by correlating IDs reported by several infected patients so as to narrow down social or local interconnections.

We propose to limit the HA's responsibility to confirming results of positively tested individuals and thus minimize the amount of data a centralized actor can derive from the protocol. All other steps are decentralized by distributing work between the users of the contact tracing system using peer-to-peer technology. Our system CAUDHT is adaptable to other popular proposals for contact tracing systems, namely PEPP-PT. We intend to provide an extension to such systems and not a replacement. To explain CAUDHT itself, we will first introduce a few important concepts.

A. Distributed Hash Tables

Conventional Bluetooth contact tracing schemes require servers run by a single entity to store IDs of at-risk persons. To replace this central instance and reduce privacy risks we propose to use a *Distributed Hash Table* (DHT). Control and operation of the storage is shifted from the HA toward the entirety of the user base. DHTs like Chord [21] or Kademlia [22] can be operated by a set of Internet-connected *nodes* and are stable even in cases of nodes leaving and joining in the system. In CAUDHT, every participating user also acts as a node in the DHT. Storage is provided in the form of a key-value-store with the ability for every participating node to store and retrieve data. Our DHT will act as a "postbox" for users. Infected individuals can store information about their health status in the DHT using the observed ID of a past contact as key. Each user periodically queries the database using their past IDs as keys. The DHT can retrieve data for a key if a message has been placed there. This mechanism allows to inform users about possible infections without a central authority contacting them.

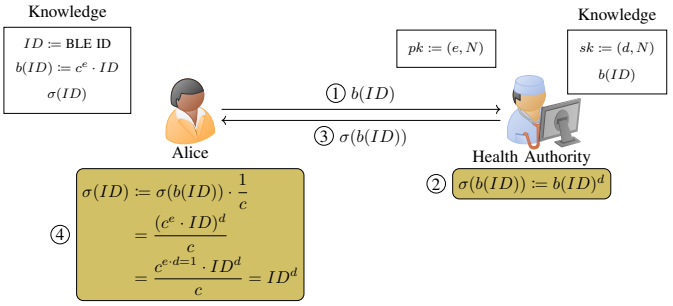

Fig. 1. An example exchange leading to a blind signature using textbook RSA. Infected patient Alice wants to retrieve a signature for an ID to prove the fact that she is infected. After blinding ID, Alice sends the blinded value to the HA who signs it without learning ID. The signature of the blinded ID is returned, which can be unblinded only by Alice, who recovers a valid signature for ID, that is also unknown to the HA. The boxes labeled *Knowledge* show all information that both parties have learned after following the protocol. Note that the HA knows neither ID, nor its signature $\sigma(ID)$ but Alice now holds a valid signature.

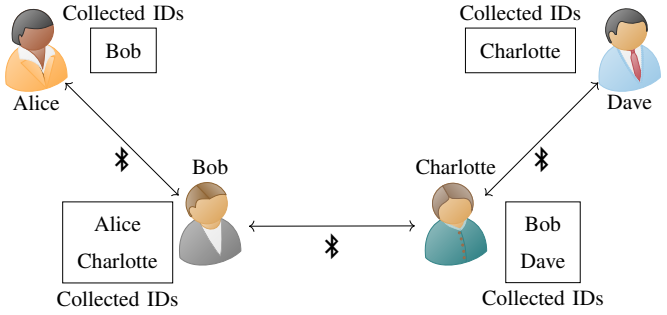

Fig. 2. During contact collection, each user stores the IDs of all devices that are in proximity. These IDs can be used to notify close contacts in case of a subsequently detected infection.

B. Blind Signatures

Postboxes can be accessed and entries can be added into by any user of the system. To prevent users from self-reporting as infected without being a verified case, they need a confirmation from the HA. Otherwise, malicious users could cause panic by leading large groups of people to believe to be at risk even though no real contact with an infected person has occurred.

On the other hand, it is important that the HA does not learn any collected ID when confirming a patient's infection. To ensure this, we use *blind signatures* [23] as a mechanism for the HA to publicly verify a users infection status. Blind signatures allow for a signer to sign a message without knowing its content but still generating a valid signature. A run of the protocol with RSA [24] as the underlying cryptographic protocol can be seen in Figure 1. The HA needs to publish her public key $pk = (e, N)$ before the protocol is run. The HA's secret key $sk = (d, N)$ remains secret.

After being tested positively for the disease, Alice wants to retrieve the HA's signature for every BLE ID ID she used during the collection phase. To do so the following steps are necessary:

① Alice transmits the *blinded* ID $b(ID)$ to the HA, masking it for the HA. Blinding is achieved by multiplying the secret value with a random number c to the power of e, a part of the HA's public key.

② The HA calculates the signature for this value using the RSA signature algorithm. In textbook RSA that signature is generated by calculating the power of d, which is a component of the HA's private key. So the signature $\sigma(b(ID))$ of the blinded ID is $(b(ID))^d$

③ The HA then transmits the signature $\sigma(b(ID))$ back to Alice.

④ Alice can unblind the message and retrieve a valid signature $\sigma(ID)$ for ID. It can be seen that Alice multiplies the message with the inverse of the blinding factor c to get the desired result.

After following the protocol the HA will only have knowledge about the previously defined secret key sk and the blinded ID, but neither ID itself, nor its signature $\sigma(ID)$. Thereby, Alice has the ability to publish these values without the HA being able to link her published IDs with Alice's identity.

C. Protocol Mechanisms

CAUDHT consists of several mechanism. A *contact collection mechanism* runs continuously on every user's end device. It collects IDs of contacts also using the system. If user Alice is tested positively, she announce her infection status to the system using the *publication mechanism*. For this purpose she retrieves signatures for seen IDs from the HA and publishes messages for the respective user at the corresponding location in the distributed database. Users regularly request their own status (i, e., if they have been in contact with an infected person) using the *polling procedure*. All processes should run independently at different frequencies.

1) Contact Collection Mechanism: To get reliable information about contacts with infected users, it is necessary to monitor the surroundings for other users and collect IDs. In a decentralized system, this step needs to be executed locally on the user's device so that no central authority learns about users' contacts. As seen in the related work, the most promising approach for this purpose is contact tracing with Bluetooth Low Energy (BLE). With BLE it is possible to scan for nearby devices. Each device can advertise an ID, which is stored when picking up a signal from that device. Due to size constraints, a sensor scan is required to retrieve the full 256 bit ID, but this scan is done automatically for BLE devices in Android and iOS [25]. By that, every user builds up a list of past contacts from recorded BLE IDs. To prevent an attacker from linking another user's locations over time, it is required to renew the own ID periodically after a certain time.

In contrast to existing systems, we propose to generate IDs from an asymmetric key pair. Our system generates an asymmetric key pair and store the secret key on the device. The public key pk_u will be used as BLE ID and broadcasted to everyone in close proximity. Other users close by record pk_u and store it as contact in their local history. Simultaneously,

the system has collected a set of public keys $pk_1, \cdots, pk_n$. This key exchange is used later on to verify that a contact with an infected person has indeed occurred.

2) Publication Mechanism: To combat COVID-19 effectively, an infected user needs to spread the news quickly to all contacts they met while being contagious (maximum the last 14 days). In order to not reveal her contact (and by that location) history to the HA, the infected user Alice does not provide her stored BLE IDs to the HA. Instead, Alice *blinds* each of the IDs she used in the last two weeks by multiplying the ID with a unique random number c^e. Let Bob be one of the respective contacts from Alice's history. Alice transmits her own blinded ID $b(ID_{Alice})$to the HA, who signs it with their public key. This key is universally known and has to be accessible by everyone. The HA then returns the corresponding signature $\sigma(b(ID_{Alice}))$ back to Alice who *unblinds* it. Alice now holds her own signed BLE ID $\sigma(ID_{Alice})$. Since each BLE ID is a public key, Alice encrypts her own BLE ID that she advertised during the last time she encountered with this specific ID of Bob. This gives her an encrypted BLE ID (EBI).

To immediately notify Bob, Alice accesses the DHT. Alice stores the signatures of Bob's EBIs at the DHT addresses corresponding to Bob's BLE ID. For example, if she encountered BLE ID ID_b while her own BLE ID was ID_a, she will store the value of $\{ID_a|\sigma(ID_a)\}_{ID_b}$ as value[1] at key position ID_b. Alice will notify all her other contacts from the time period when she was contiguous following this pattern.

This approach works like a postbox service, where each user can get messages delivered by polling their own previous BLE IDs in the DHT. So Bob will only learn that he is possibly infected by searching for his past BLE ID ID_b. It is important that potential contacts are warned quickly in case of a confirmed infection. Therefore, every user should poll their postboxes at least once every few hours. Figure 3 shows how Alice publishes her infection status in the DHT so that Bob can learn that he is at risk.

[1] $\{X\}_K$ means that X is encrypted using public key K.

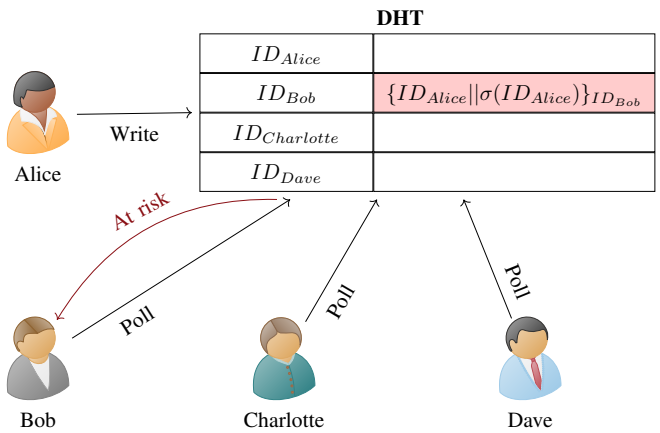

Fig. 3. An example for the publication and polling mechanism. Alice publishes her infection status to her previous contact Bob, who can retrieve this new information by polling for his own ID in the DHT.

3) Polling Mechanism: To understand if a user has been in contact with a person who was recently infected, they need to query the DHT periodically. Assume that curious user Bob met Alice recently and used the BLE ID ID_b at that time. Since then Alice has been diagnosed and left a message in Bob's postbox at this specific key. If Bob now performs a search for this key in the DHT an encrypted result will be returned. Because he used ID_b as his BLE ID at one point, Bob can decrypt Alice's message using the corresponding private key. This gives him Alice's BLE ID ID_a as well as the signature from the HA regarding this BLE ID. The signature confirms to him that Alice's test result were indeed positive. By looking up Alice's BLE ID in his own history he can also confirm that he has indeed encountered Alice in the past. Without this lookup, a malicious positively tested patient Eve could claim to have seen many BLE IDs resulting random users into believing they have contracted the disease.

V. DISCUSSION

CAUDHT provides security against several attack vectors that were identified in the attacker model in Section III. Defense mechanisms against various attack vectors are discussed in the following.

A. Health Authority

The HA is not able to learn anything about infected patients' contact histories. Even if several infected patients have seen the same BLE ID, the HA will not be able to link them together because these values are not transmitted. Observing the DHT does not leak additional information to the HA, assuming the number of infected patients is large enough so that timing correlations of DHT write operations are masked by a steady stream of updates from multiple infected parties.

B. Infected Users

An infected user is not able to spread panic and misinformation as users check their local contact history for the infected patient's BLE ID for validation. A non-infected person Eve can not claim to be infected since user will not accept a message lacking the HA's signature. That signature is only provided for people that have tested positive for the disease.

C. Uninfected Users

A user Bob learns the BLE ID of the infected patient Alice, when encrypting his messages. So he will know that a user with this ID is now sick. This information is leaked intentionally so Bob can check if a contact with Alice was indeed recorded or if this is an attempt to spread panic. This trade-off can be reversed by not providing the infected patient's BLE ID in the message to the user's postbox.

The DHT is operated by all system users. A malicious participant Eve could request BLE IDs she has seen, however, only a user holding the private key to the IDs can decrypt the message. Even though Eve cannot decrypt an answer

message, the fact that a message was returned can already leak information. If messages are only placed in the DHT when an infection is confirmed, Eve can conclude that the holder of the requested BLE ID has been in contact with an infected individual. To prevent this, postboxes can hold more than one message and users occasionally write random messages into their own (or other user's) postboxes. Such messages will not contain readable content or a signature from the HA and will be discarded by the recipient. This way it is not possible to determine the infection status by requesting an entry from the DHT.

D. Security Enhancements by using a DHT and Blind Signatures

Both the DHT and blind signatures solve different security problems in our decentralized design. First, the DHT solves the problem of distributing the data about infections. Theoretically, a central database could be accessed using anonymizing proxies like Tor [26] to hide the requesting user's identity. However, systems like Tor are not as scalable as a DHT where very user participates as a node automatically.

In contrast to a centralized approach where infection messages are provided by a database (or broadcasts) operated by a third party such as the HA, our system relies on infected patients messaging their contacts. To prevent misinformation about infection statuses the blind signatures ensure that only infected patients are able to inform their contacts about an infection.

These two building blocks are necessary to operate or proposed trustworthy and scalable decentralized contact tracing system.

E. Scalability

When evaluating a decentralized algorithm, it is always important to consider if a large-scale installation of the system can still run efficiently. Additional concern should be laid on data traffic created by the DHT. Many users will interact with the DHT while on metered mobile connections.

To ensure that the DHT does not overflow with outdated data, entries need to be deleted once they are no longer useful. Because contact information is only interesting for 14–21 days in case of COVID-19, entries that are older than this time can and should be deleted. Each DHT node ensures that all values stored at keys for which it is responsible are up to date. This can be achieved by adding a timestamp to each message specifying when it can be safely deleted. DHT values (i. e., postbox messages) bearing timestamps older than three weeks are definitely not used anymore and can be discarded.

Growth of the DHT itself is no major scalability problem. Each new potential postbox user is also part of the DHT's set of nodes and helps storing the data. In the long term, the amount of data stored per user is constant regardless of the number of participants.

VI. CONCLUSION AND FUTURE WORK

In this whitepaper we introduced several privacy-preserving additions to Bluetooth-based contact tracing approaches for COVID-19. Our main contributions are:

- Using blind signatures for allowing the infection status of an ID to be verifiable while keeping the HA from learning private information.
- The introduction of a distributed approach to contact tracing where only the disease testing is conducted by a central instance.
- A DHT-based postbox system where users can communicate directly with each other.
- Defense against different attack vectors, including malicious actors that target on spreading panic and misinformation.

Future work amounts to further evaluate our ideas, to fully implement them, and to potentially collaborate with the community operating such contact tracing apps. The success of contact tracing apps relies on cooperation by the population, so we are convinced that proper user education about why and how their private data is protected is a key element in fighting this disease.

REFERENCES

[1] Hamilton, Isobel Asher, "11 countries are now using people's phones to track the coronavirus pandemic, and it heralds a massive increase in surveillance ," accessed: 26. March 2020. [Online]. Available: www.businessinsider.com/countries-tracking-citizens-phones-coronavirus-2020-3?r=DE&IR=T

[2] Chaos Computer Club, "10 requirements for the evaluation of "contact tracing" apps." [Online]. Available: https://www.ccc.de/en/updates/2020/contact-tracing-requirements

[3] K. T. Eames and M. J. Keeling, "Contact tracing and disease control," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 270, no. 1533, pp. 2565–2571, 2003.

[4] G. Webb, C. Browne, X. Huo, O. Seydi, M. Seydi, and P. Magal, "A model of the 2014 ebola epidemic in west africa with contact tracing," *PLoS currents*, vol. 7, 2015.

[5] R. Huerta and L. S. Tsimring, "Contact tracing and epidemics control in social networks," *Physical Review E*, vol. 66, no. 5, p. 056115, 2002.

[6] "Germany Increases Coronavirus Threat to "High"," accessed: 05. April 2020. [Online]. Available: www.spiegel.de/international/germany/germany-increases-coronavirus-threat-to-high-a-a8fa63e2-2123-4c8c-aa73-f557244aaf07

[7] T. Altuwaiyan, M. Hadian, and X. Liang, "Epic: Efficient privacy-preserving contact tracing for infection detection," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[8] Government of Singapore, "TraceTogether," last access: 06. April 2020. [Online]. Available: www.tracetogether.gov.sg

[9] H. Cho, D. Ippolito, and Y. W. Yu, "Contact tracing mobile apps for covid-19: Privacy considerations and related trade-offs," *arXiv preprint arXiv:2003.11511*, 2020.

[10] ePP-PT e.V. i.Gr, "PePP-PT," last access: 05. April 2020. [Online]. Available: www.pepp-pt.org

[11] Troncoso, Carmela and Payer, Mathias and Hubaux, Jean-Pierre and Salathé, Marcel and Larus, James and Bugnion, Edouard and Lueks, Wouter and Stadler, Theresa and Pyrgelis, Apostolos and Antonioli, Daniele and Barman, Ludovic and Chatel, Sylvain and Paterson, Kenneth and Capkun, Srdjan and Basin, David and Jackson, Dennis and Preneel, Bart and Smart, Nigel and Singelee, Dave and Abidin, Aysajan and Guerses, Seda and Veale, Michael and Cremers, Cas and Binns, Reuben and Wiegand, Thomas , "Decentralized Privacy-Preserving Proximity Tracing," last access: 08. April 2020. [Online]. Available: https://github.com/DP-3T/documents

[12] Massachusetts Institute of Technology, " Private Kit: Safe Paths; Privacy-by-Design Contact Tracing ," last access: 06. April 2020. [Online]. Available: safepaths.mit.edu

[13] R. Raskar, I. Schunemann, R. Barbar, K. Vilcans, J. Gray, P. Vepakomma, S. Kapa, A. Nuzzo, R. Gupta, A. Berke *et al.*, "Apps gone rogue: Maintaining personal privacy in an epidemic," *arXiv preprint arXiv:2003.08567*, 2020.

[14] Covid Watch, "Covid Watch," last access: 07. April 2020. [Online]. Available: www.covid-watch.org

[15] "Indian government releases COVID-19 contact tracing app," accessed: 06. April 2020. [Online]. Available: www.medianama.com/2020/03/223-government-apps-coronavirus

[16] Austrian Red Cross, " Stopp Corona – Mein Kontakt-Tagebuch ," last access: 06. April 2020. [Online]. Available: www.roteskreuz.at/site/faq-app-stopp-corona/

[17] The New York Times, "To Track Coronavirus, Israel Moves to Tap Secret Trove of Cellphone Data," last access: 06. April 2020. [Online]. Available: www.nytimes.com/2020/03/16/world/middleeast/israel-coronavirus-cellphone-tracking.html?referringSource=articleShare

[18] L. Reichert, S. Brack, and B. Scheuermann, "Privacy-preserving contact tracing of covid-19 patients," Cryptology ePrint Archive, Report 2020/375, 2020. [Online]. Available: https://eprint.iacr.org/2020/375

[19] A. Berke, M. Bakker, P. Vepakomma, R. Raskar, K. Larson, and A. Pentland, "Assessing disease exposure risk with location histories and protecting privacy: A cryptographic approach in response to a global pandemic," *arXiv preprint arXiv:2003.14412*, 2020.

[20] S. Jeong, S. Kuk, and H. Kim, "A smartphone magnetometer-based diagnostic test for automatic contact tracing in infectious disease epidemics," *IEEE Access*, vol. 7, pp. 20 734–20 747, 2019.

[21] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, 2001.

[22] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.

[23] D. Chaum, "Blind signatures for untraceable payments," in *Advances in cryptology*. Springer, 1983, pp. 199–203.

[24] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[25] Argenox Technologies LLC., "BLE Advertising Primer," last access: 05. April 2020. [Online]. Available: www.argenox.com/library/bluetooth-low-energy/ble-advertising-primer

[26] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Naval Research Lab Washington DC, Tech. Rep., 2004.