P^{2}B-Trace: Privacy-Preserving Blockchain-based Contact Tracing to Combat Pandemics

Zhe Peng
Hong Kong Baptist University
pengzhe@comp.hkbu.edu.hk

Cheng Xu
Hong Kong Baptist University
chengxu@comp.hkbu.edu.hk

Haixin Wang
Hong Kong Baptist University
hxwang@comp.hkbu.edu.hk

Jinbin Huang
Hong Kong Baptist University
jbhuang@comp.hkbu.edu.hk

Jianliang Xu
Hong Kong Baptist University
xujl@comp.hkbu.edu.hk

Xiaowen Chu
Hong Kong Baptist University
chxw@comp.hkbu.edu.hk

ABSTRACT

The eruption of a pandemic, such as COVID-19, can cause an unprecedented global crisis. Contact tracing, as a pillar of communicable disease control in public health for decades, has shown its effectiveness on pandemic control. Despite intensive research on contact tracing, existing schemes are vulnerable to attacks and can hardly simultaneously meet the requirements of data integrity and user privacy. The design of a privacy-preserving contact tracing framework to ensure the integrity of the tracing procedure has not been sufficiently studied and remains a challenge. In this paper, we propose P^{2}B-Trace, a privacy-preserving contact tracing initiative based on blockchain. First, we design a decentralized architecture with blockchain to record an authenticated data structure of the user's contact records, which prevents the user from intentionally modifying his local records afterward. Second, we develop a zero-knowledge proximity verification scheme to further verify the user's proximity claim while protecting user privacy. We implement P^{2}B-Trace and conduct experiments to evaluate the cost of privacy-preserving tracing integrity verification. The evaluation results demonstrate the effectiveness of our proposed system.

KEYWORDS

Contact tracing; Integrity; Privacy-preserving; Blockchain

ACM Reference Format:
Zhe Peng, Cheng Xu, Haixin Wang, Jinbin Huang, Jianliang Xu, and Xiaowen Chu. 2021. P^{2}B-Trace: Privacy-Preserving Blockchain-based Contact Tracing to Combat Pandemics. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21), June 20–25, 2021, Virtual Event, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3448016.3459237

1 INTRODUCTION

Large-scale pandemic outbreaks have devastating effects on the health and well-being of the global population [16]. Take COVID-19 as an example, as of 23rd March 2021, there have been 123,627,191 cases and 2,722,203 deaths confirmed across 192 countries and regions [11]. The eruption of the pandemic has carried intense pressures on both healthcare professionals (e.g., doctors and nurses) and healthcare systems. In response to this unprecedented global crisis, many countries have implemented non-pharmaceutical interventions (NPI) [10] to reduce the transmission of the virus, including the closure of workplaces, schools, and national lockdowns.

Contact tracing, a type of social distancing measure, aims to identify and track the people who have come into contact with an infected person. Identified close contacts can be provided with early quarantine, screening, diagnosis, and treatment to break the virus's transmission chain. Most governments across the globe have leveraged this tracking strategy to ease the rigorous social distancing restrictions. Specifically, the experience in HK has indicated that the spread of COVID-19 can be effectively contained with the help of contact tracing by reducing the community transmission from unidentified cases [8]. However, traditional manual contact tracing is inefficient because it is limited by a person's ability to recall all the close contacts and the time it takes to reach these contacts. Thus, in a world coexisting with the infectious coronavirus, an effective and secure digital contact tracing system is much desired. This system could help to rescue the economy while saving lives and resuming the normality, especially when the lockdown is lifted (or partially lifted) and the society steps into a "new normal" [7].

There are several challenges in building an effective digital contact tracing system. First, *user privacy* protection, especially during contact data collection and processing, is known to have a significant impact on the uptake of such a system [23]. Thus, safeguarding user privacy should be the very first requirement for digital contact tracing. Most existing contact tracing systems [3, 20, 21] are designed with a centralized model, where personal data is uploaded to a central server for contact matching. However, in such systems, user privacy could be compromised and system security is not guaranteed. With the collected personal data (e.g., identities, locations, etc.), the server might infer knowledge pertaining to users' interests. Moreover, the server constitutes a valuable target for malicious attackers, which may result in serious data breach and leakage.

Second, verifying *data integrity* is crucial to ensure correct contact matching and proximity claim. This challenge is two-fold: (i) how to prevent confirmed patients from modifying their historical contact records to generate false matching; and (ii) how to prevent unconfirmed clients from tampering with matching results to generate false exposure claim. Prior systems [14, 19, 28] mainly utilize Bluetooth to determine the relative distances among users to

SIGMOD '21, June 20–25, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8343-1/21/06…$15.00
https://doi.org/10.1145/3448016.3459237

enable decentralized contract tracing. In the decentralized model, data will be collected and stored on the user's local device, which gives more control back to the user. However, most approaches can hardly guarantee the integrity of the user's local contact data, leaving space for data fraud. For example, an infected person might report fraudulent contact records to cause unnecessary public panics. Unconfirmed users may modify their contact data to change matching results for some benefits (e.g., free screening and material subsidies) from the authority or avoiding quarantine.

Contributions. To meet these challenges, we propose P^{2}B-Trace, a blockchain-based contact tracing system, which simultaneously ensures data integrity and user privacy. This system enables users to confidentially conduct contact matching on local devices based on historical contact records, where a trusted server is unnecessary. Concretely, we make the following major contributions.

- We design a decentralized architecture with blockchain to record an authenticated data structure (ADS) of the user's contact records, which prevents the user from intentionally modifying his records afterward. In addition, we develop a zero-knowledge proximity verification scheme to verify the user's proximity claim while protecting user privacy.
- We implement P^{2}B-Trace using readily-available infrastructural primitives to demonstrate practicability. Empirical results show that P^{2}B-Trace achieves a verifiable and privacy-preserving contact tracing system in a decentralized manner. Moreover, proof generation time and verification time are identified as the cost-critical parts of privacy-preservation.
- We discuss future research directions and challenges in data management created by the blockchain-based contact tracing application. Some potential solutions are also presented.

2 PROBLEM DEFINITION

System Model. P^{2}B-Trace is a decentralized system for verifiable privacy-preserving contact tracing. The system will notify users after contact matching on local mobile devices if they have been exposed to an infected person. Figure 1 presents the P^{2}B-Trace system architecture. Our system mainly consists of four actors: (i) *patient*, (ii) *client*, (iii) *worker*, and (iv) *authority*. The patient is a virus-infected person, while the client refers to an unconfirmed person. The worker is responsible for constructing the consensus proofs and appending new blocks to the blockchain [5, 12]. The authority represents a component in the healthcare system (e.g., a hospital or a government sector), which can provide a close proximity certification for further screening.

Compatible with most decentralized contact tracing systems, P^{2}B-Trace also uses Bluetooth to generate and exchange anonymous user-oriented identifiers with smartphones. Users update their anonymous identifiers at regular intervals (e.g., 15 minutes) and constantly record encountered identifiers nearby. As such, each user builds (i) a *Generation List* $\mathcal{L}_{\mathcal{G}} = \{GID_1, \cdots, GID_n\}$, where GID_i is a generated identifier at ith interval, and (ii) an *Encounter List* $\mathcal{L}_{\mathcal{E}} = \{EID_1, \cdots, EID_m\}$, where EID_i is an encountered identifier. The two lists can be built at a specific frequency (e.g., every day). Moreover, to enable verifiable contact tracing, an *authenticated data structure* (ADS) is constructed for each $\mathcal{L}_{\mathcal{G}}$ and $\mathcal{L}_{\mathcal{E}}$ (denoted as $\mathcal{R}_{\mathcal{G}}$ and $\mathcal{R}_{\mathcal{E}}$, respectively) and stored on the blockchain.

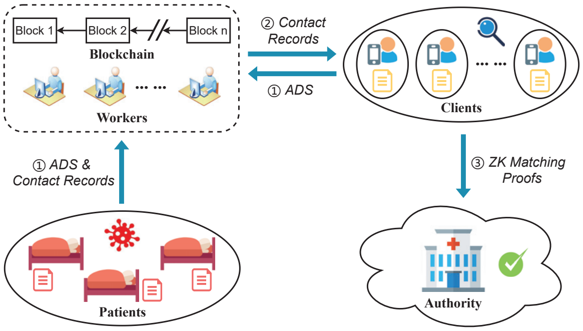

Figure 1: The system architecture of P^{2}B-Trace.

To enable contact tracing, once confirmed with the infection, the patient uploads historical generation lists during a tracing period (normally within the past 14 days) to the blockchain. Observing the patient's contact records, all clients conduct a cross-check with respective encounter lists on local mobile devices. If a match is found, the client will be notified and s/he could report this case to the authority for further follow-up. Specifically, to ensure data integrity in the tracing procedure, the verification will be conducted in two aspects. First, the integrity of the patient's contact records is verified with the corresponding ADS stored on the blockchain such that no fraudulent data are reported. Second, to simultaneously prevent false proximity claims and safeguard user privacy, the client needs to generate zero-knowledge proof for the authority such that the contact matching can be verified without privacy exposure. Moreover, in the whole process, both data storage and data computation are realized in a decentralized manner to protect user privacy and make the system more scalable.

Threat Model. Three security threats are considered in this paper: (i) the patient is not fully trusted and might upload falsified or incomplete contact records when confirmed with the pandemic; (ii) the client is not fully trusted and might tamper with matching results to generate false exposure claim; and (iii) the authority is curious and may attempt to obtain the knowledge pertaining to its interests from the received data.

To address the first threat, we adopt verifiable contact report to ensure the integrity of the patient's contact records. Specifically, a smart contract reconstructs an ADS from the uploaded generation list $\mathcal{L}_{\mathcal{G}}$, and compares it with the corresponding ADS stored on the blockchain. Using the ADS pinned on the blockchain, the system can establish the soundness and completeness of the generation list $\mathcal{L}_{\mathcal{G}}$ uploaded by the patient: (i) *Soundness.* No generated identifiers in $\mathcal{L}_{\mathcal{G}}$ are tampered with and all of them satisfy the temporal conditions. (ii) *Completeness.* No valid generated identifier is missing regarding the tracing period.

To further address the second and the third threat, we advocate privacy-preserving verifiable proximity detection that enables the client to prove the correctness of the matching results without revealing privacy. We define three levels of security requirements:

- **Generation Inclusiveness.** The matched identifier is included in a generation list $\mathcal{L}_{\mathcal{G}}$ uploaded by the patient. That is, the patient has generated the matched identifier during the tracing period.

- **Encounter Inclusiveness.** In addition to generation inclusiveness, the matched identifier is also included in the client's encounter list $\mathcal{L}_\mathcal{E}$. That is, the client has been exposed to a patient and collected the patient's generated identifier.
- **Zero-Knowledge Confidentiality.** Any information beyond the fact that an identifier is included in both $\mathcal{L}_\mathcal{G}$ and $\mathcal{L}_\mathcal{E}$ is protected. That is, the authority can gain nothing about $\mathcal{L}_\mathcal{E}$ (e.g., not even the matched identifier or list size).

We assume that there is no collusion between patients, clients, and the authority. In addition, following the standard threat model in existing blockchain systems [9, 22, 25], we consider a blockchain system of N workers under the BFT fault model, where at most $f = \frac{N-1}{3}$ workers could be malicious.

3 THE P^{2}B-TRACE FRAMEWORK

In this section, we present the P^{2}B-Trace, a privacy-preserving contact tracing initiative based on blockchain. We begin by introducing a contact data verification scheme to prevent the patient's local contact records fraud. Furthermore, we propose a zero-knowledge proximity verification scheme to prevent the client from tampering with contact matching results while preserving user privacy.

ADS Generation and Storage. The verification for the tracing procedure is built upon a blockchain-based hybrid storage model. Recall that in the proposed P^{2}B-Trace architecture, a user periodically generates a new anonymous identifier and constantly collects adjacent users' identifiers through the Bluetooth module. As such, in each period (e.g., one day), each user can build a generation list $\mathcal{L}_\mathcal{G}$ and an encounter list $\mathcal{L}_\mathcal{E}$. During the local data storage, the user also constructs an *authenticated data structures* (ADS) for each contact list (denoted as $\mathcal{R}_\mathcal{G}$ and $\mathcal{R}_\mathcal{E}$ for $\mathcal{L}_\mathcal{G}$ and $\mathcal{L}_\mathcal{E}$, respectively). Concretely, we utilize Merkle Hash Tree [18] to derive a Merkle Root (serving as the ADS) to support the data integrity verification.

Emerging Blockchain technology has been adopted as a trusted data storage solution in various fields [13, 24, 26]. In P^{2}B-Trace, both $\mathcal{R}_\mathcal{G}$ and $\mathcal{R}_\mathcal{E}$ are stored on the blockchain as notarization of the original contact records. The benefits obtained from the blockchain are two-fold. First, the blockchain is maintained by untrusted peers in a decentralized P2P network, which keeps the decentralized architecture of our system. Second, the on-chain ADS is immutable, which can be used to verify the integrity of the patient's uploaded contact records and the client's claimed close proximities. Figure 2 illustrates the data hybrid-storage based on blockchain.

Contact Data Verification. In the tracing procedure, if a user is confirmed with the pandemic, this patient needs to upload his/her historical generation lists $\mathcal{L}_\mathcal{G}$ during a tracing period (normally within the past 14 days) for contact matching. In P^{2}B-Trace, the patient will upload the data to the blockchain. To be noticed, some variations could be made for the data uploading manner in practical scenarios, which do not change the fundamental nature of our system. For example, the patient can broadcast the data through a trusted third-party server or a self-owned website.

In order to ensure the integrity of the patient's uploaded generation lists $\mathcal{L}_\mathcal{G}$, a smart contract is deployed on the blockchain to verify the contact data. The ADS is reconstructed from $\mathcal{L}_\mathcal{G}$ and compared with the corresponding ADS stored on the blockchain. As such, the soundness and completeness of the patient's uploaded contact records can be successfully verified.

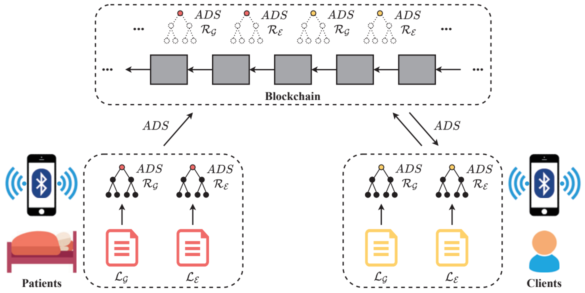
ADS
$\mathcal{R}_\mathcal{G}$
ADS
$\mathcal{R}_\mathcal{E}$
ADS
$\mathcal{R}_\mathcal{G}$
ADS
$\mathcal{R}_\mathcal{E}$
Blockchain
ADS
ADS
ADS
$\mathcal{R}_\mathcal{G}$
ADS
$\mathcal{R}_\mathcal{E}$
ADS
$\mathcal{R}_\mathcal{G}$
ADS
$\mathcal{R}_\mathcal{E}$
Patients
$\mathcal{L}_\mathcal{G}$
$\mathcal{L}_\mathcal{E}$
$\mathcal{L}_\mathcal{G}$
$\mathcal{L}_\mathcal{E}$
Clients

Figure 2: Data hybrid-storage based on blockchain.

Zero-Knowledge Proximity Verification. With the verified patient's contact records, clients can conduct a cross-check with respective encounter lists $\mathcal{L}_\mathcal{E}$ on local mobile devices. If a match is found, the client will be notified and s/he could report this case to the authority for further follow-up. However, the client might modify local encounter records to change the contact matching result for some benefits (e.g., free screening and material subsidies) from the authority or avoiding quarantine.

A basic solution to the client's proximity verification is to use joint inclusiveness verification based on the ADS stored on the blockchain. More specifically, for any successful contact match, the client generates a verification object ($\mathcal{VO}_\mathcal{G}$) from the patient's $\mathcal{R}_\mathcal{G}$ and a verification object ($\mathcal{VO}_\mathcal{E}$) from the client's $\mathcal{R}_\mathcal{E}$. The matched identifier along with the $\mathcal{VO}_\mathcal{G}$ and $\mathcal{VO}_\mathcal{E}$, are returned to the authority for verifying the inclusiveness in both the patient's $\mathcal{L}_\mathcal{G}$ and the client's $\mathcal{L}_\mathcal{E}$. However, this solution will undermine the client's privacy by revealing the matched identifier and the generated $\mathcal{VO}_\mathcal{G}$ and $\mathcal{VO}_\mathcal{E}$. This violates our zero-knowledge confidentiality requirement as mentioned before.

Thus, we develop a zero-knowledge proximity verification scheme based on Bulletproofs [6], which prevents the client from tampering with local contact matching results while preserving user privacy.

Definition 3.1 (Zero-Knowledge Proximity Verification). For the input identifier *id*, ADS $\mathcal{R}_\mathcal{G}$ of $\mathcal{L}_\mathcal{G}$, ADS $\mathcal{R}_\mathcal{E}$ of $\mathcal{L}_\mathcal{E}$, and global parameters $G, H \in E(F_q)$, this scheme can prove to a verifier $\mathcal{V}$ that the prover $\mathcal{P}$ knows an assignment to *id* such that $id \in \mathcal{L}_\mathcal{G} \land id \in \mathcal{L}_\mathcal{E}$, without revealing *id*. It consists of the following algorithms:

- $\{\mathbf{G} = (G_1, \cdots, G_n), \mathbf{H} = (H_1, \cdots, H_n), G, H\} \leftarrow \mathsf{Setup}(1^\lambda)$: On input a security parameter 1^λ, it outputs public parameters $\{\mathbf{G}, \mathbf{H}, G, H\}$ acting as implicit input for other functions.
- $\pi \leftarrow \mathsf{ProofGen}(id, \mathcal{R}_\mathcal{G}, \mathcal{R}_\mathcal{E}, \mathbf{r})$: On input an identifier *id*, an ADS $\mathcal{R}_\mathcal{G}$, an ADS $\mathcal{R}_\mathcal{E}$, and random parameters $\mathbf{r}$, it outputs a zero-knowledge proximity proof π.
- $\{0, 1\} \leftarrow \mathsf{ProofVer}(\pi, \mathcal{R}_\mathcal{G}, \mathcal{R}_\mathcal{E})$: On input a zero-knowledge proximity proof π, an ADS $\mathcal{R}_\mathcal{G}$, and an ADS $\mathcal{R}_\mathcal{E}$, it outputs 1 if the verification is valid.

To efficiently implement the cryptographic hash function (such as SHA-256) in arithmetic circuits (AC), an optimization method from prior work [27] is utilized to improve the computational efficiency. Thus, benefitted from the property of no trusted setup in Bulletproofs, our zero-knowledge proximity verification scheme

Table 1: Data Storage Performance (KB/s)

Method	# daily confirmed cases ($\times 10^4$)						
	1	5	10	15	20	25	30
BeepTrace	222	851	673	574	483	426	351
P^{2}B-Trace	115	410	983	746	698	558	571

can effectively verify the client’s contact matching results while safeguarding user privacy in a decentralized manner.

4 EXPERIMENTAL EVALUATION

Setup. We use Hyperledger Fabric to set up the blockchain for its adaptability. The smart contract for storing data on the blockchain and the zero-knowledge proximity verification with SHA-256 are implemented with Golang and Python, respectively. We use M191 as the elliptic curve. Experiments are conducted on a desktop computer with a 3.2 GHz quad-core Intel Xeon processor and 64 GB RAM.

Performance Evaluation. We first compare data storage performance with BeepTrace [30] by varying the number of daily confirmed cases. Table 1 reports the system throughput when confirmed users send transactions containing their data records to the blockchain. It is shown that P^{2}B-Trace can achieve higher efficiency for more daily confirmed cases. The reason is that workers in BeepTrace need to maintain two blockchains simultaneously.

We further evaluate the integrity verification scheme with three metrics: (i) size of the verification proof, (ii) proof generation cost in terms of prover CPU time, and (iii) proof verification cost in terms of verifier CPU time. Four methods are compared: only patient’s data verification (DV), only client’s zero-knowledge proximity verification (ZKPV), only client’s proximity verification without zero-knowledge (PV), and our P^{2}B-Trace verification scheme.

Figure 3 shows the results when the number of identifiers is varied from 2^1 to 2^7. Specifically, the amount $2^7 = 128$ is sufficient. Since in a typical scenario where the user updates an identifier every 15 minutes and generates a new list every day (14 active hours for an adult), the total amount of identifiers is 56. It can be seen that the proof size keeps succinct (< 10 KB) in P^{2}B-Trace. For the patient’s data integrity verification, benefitted from the constructed ADS, both proof generation and verification are efficient with short execution time. For the client’s proximity verification, the time-oriented metrics increase noteworthily under P^{2}B-Trace, but those of the approach without zero-knowledge fluctuate a bit. The reason is the AC-based implementation takes more time for integrity verification with privacy protection. Moreover, Bulletproofs provides smaller proofs at the expense of computational cost.

Discussion. Our results show that the Bulletproofs used in P^{2}B-Trace have an effect on the verification efficiency as the dominant factor. To improve the verification efficiency, some alternatives such as SNARKs [4] and SNORKs [17], can be used for zero-knowledge proof. However, these schemes need a trusted setup and the proof size is larger. In addition, the implementation-specific optimization (such as parallelization) is also worthy to be sustainably explored.

5 FUTURE DIRECTIONS

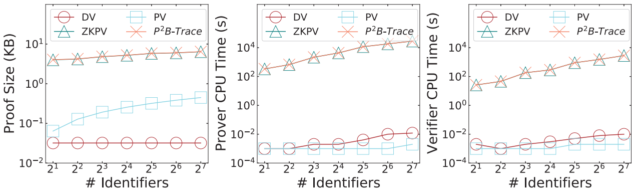

Figure 3: Integrity Verification Performance.

System Scalability. To enable verifiable contact tracing, the ADSs and contact data are stored on blockchain. The key challenge is how to efficiently store data for the users in different countries or even in the globe. To tackle this issue, a hierarchical chain can be first built for the ADS by increasing the tree depth and reducing the update frequency. The ADS stored in the upper chain is constructed from the ADS stored in the lower chain with a lower frequency. Second, we can explore the emerging sharding [9] technique to dynamically maintain regional sub-chains to improve scalability. Moreover, a data compression scheme and a hybrid-storage model to outsource data can be studied to reduce the on-chain data size.

Data Query. In the contact data verification stage, smart contract needs to query the corresponding on-chain ADS. The key issue is how to efficiently retrieve data on the blockchain [22, 29, 32]. To solve this problem, a world state trie can be tailored and maintained in the block to store ADS within the latest 14 days for each user. Alternatively, another idea is to design a packaging strategy to build a data-oriented blockchain by storing relevant data in the same block. In addition, in decentralized contact tracing, each user needs to conduct contact matching with a large number of patients. To improve matching efficiency, an index can be periodically generated for patients’ uploaded contact records by a smart contract.

Secure Computation. During the zero-knowledge proximity verification, the performance enhancement is still a problem to be addressed. Apart from using other zero-knowledge arguments, a possible direction is to encrypt the original data and outsource the verification execution to a blockchain. Some preliminary approaches have been proposed (e.g., via homomorphic encryption [1] or multi-party computation [2, 15]). Alternatively, a trusted execution environment (TEE) [31] such as Intel SGX can be leveraged to conduct the verification on the blockchain. In such settings, novel consensus algorithms will be designed to coordinate TEE-enabled nodes and non-TEE nodes with consideration of system security.

6 CONCLUSION

In this paper, we propose P^{2}B-Trace, a privacy-preserving contact tracing initiative based on blockchain. A decentralized architecture is designed to record the ADS of contact records for preventing data modification. Then, we propose a zero-knowledge proximity verification scheme to verify proximity claim with privacy protection. We implement P^{2}B-Trace and evaluation results demonstrate the effectiveness of our proposed system.

ACKNOWLEDGMENTS

This work was supported by Hong Kong RGC projects 12201520, 12200819, and Guangdong Basic and Applied Basic Research Foundation 2020A1515111070.

REFERENCES

[1] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–35.

[2] Johes Bater, Gregory Elliott, Craig Eggen, Satyender Goel, Abel N Kho, et al. 2017. SMCQL: Secure Query Processing for Private Data Networks. In *PVLDB*.

[3] Jason Bay, Joel Kek, Alvin Tan, et al. 2020. BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders. *Government Technology Agency of Singapore Tech Rep* (2020).

[4] Eli Ben-Sasson, Alessandro Chiesa, et al. 2013. SNARKs for C: Verifying program executions succinctly and in zero knowledge. In *Proc. of CRYPTO*.

[5] Yehonatan Buchnik and Roy Friedman. 2020. FireLedger: a high throughput blockchain consensus protocol. In *PVLDB*.

[6] Benedikt Bünz, Jonathan Bootle, Dan Boneh, et al. 2018. Bulletproofs: Short proofs for confidential transactions and more. In *Proc. of IEEE SP*.

[7] Noel Carroll and Kieran Conboy. 2020. Normalising the "new normal": changing tech-driven work practices under pandemic time pressure. *International Journal of Information Management* (2020).

[8] Benjamin J Cowling, Sheikh Taslim Ali, et al. 2020. Impact assessment of non-pharmaceutical interventions against coronavirus disease 2019 and influenza in Hong Kong: an observational study. *Lancet Public Health* 5 (2020), 279–288.

[9] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. 2019. Towards scaling blockchain systems via sharding. In *Proc. of ACM SIGMOD*.

[10] Seth Flaxman, Swapnil Mishra, Axel Gandy, H Juliette T Unwin, et al. 2020. Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. *Nature* 584, 7820 (2020), 257–261.

[11] Center for Systems Science and Engineering at Johns Hopkins University. 2020. COVID-19 Dashboard. https://coronavirus.jhu.edu/map.html

[12] Shang Gao, Zecheng Li, et al. 2019. Power adjusting and bribery racing: Novel mining attacks in the bitcoin system. In *Proc. of ACM CCS*.

[13] Suyash Gupta, Sajjad Rahnama, et al. 2020. Resilientdb: Global scale resilient blockchain fabric. In *PVLDB*.

[14] Apple Inc. and Google LLC. 2020. Exposure Notifications. https://www.google.com/covid19/exposurenotifications/

[15] Marcel Keller. 2020. MP-SPDZ: A versatile framework for multi-party computation. In *Proc. of ACM CCS*.

[16] Carolina Lucas, Patrick Wong, Jon Klein, Tiago BR Castro, Julio Silva, et al. 2020. Longitudinal analyses reveal immunological misfiring in severe COVID-19. *Nature* 584, 7821 (2020), 463–469.

[17] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. 2019. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In *Proc. of ACM CCS*.

[18] Ralph C Merkle. 1989. A certified digital signature. In *CRYPTO*.

[19] Department of Health Australia. 2020. COVIDSafe. https://www.health.gov.au/resources/apps-and-tools/covidsafe-app#

[20] Gov. of India. 2020. Aarogya Setu. https://www.mygov.in/aarogya-setu-app/

[21] Government of Singapore. 2020. SafeEntry. https://www.safeentry.gov.sg

[22] Yanqing Peng, Min Du, Feifei Li, et al. 2020. FalconDB: Blockchain-based Collaborative Database. In *Proc. of ACM SIGMOD*. 637–652.

[23] Zhe Peng, Jinbin Huang, Haixin Wang, et al. 2021. BU-Trace: A Permissionless Mobile System for Privacy-Preserving Intelligent Contact Tracing. In *DASFAA*.

[24] Zhe Peng, Jianliang Xu, Xiaowen Chu, et al. 2021. VFChain: Enabling Verifiable and Auditable Federated Learning via Blockchain Systems. *IEEE Transactions on Network Science and Engineering* (2021).

[25] Chao Qiu, Haipeng Yao, F Richard Yu, Chunxiao Jiang, and Song Guo. 2019. A service-oriented permissioned blockchain for the Internet of Things. *IEEE Transactions on Services Computing* 13, 2 (2019), 203–215.

[26] Pingcheng Ruan, Gang Chen, Tien Tuan Anh Dinh, et al. 2019. Fine-grained, secure and efficient data provenance on blockchain systems. In *PVLDB*.

[27] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, et al. 2014. Zerocash: Decentralized anonymous payments from bitcoin. In *Proc. of IEEE SP*.

[28] United Kingdom National Health Service. 2020. NHS Covid-19 App. https://www.nhs.uk/using-the-nhs/nhs-services/the-nhs-app/

[29] Cheng Xu, Ce Zhang, and Jianliang Xu. 2019. vChain: Enabling verifiable boolean range queries over blockchain databases. In *Proc. of ACM SIGMOD*.

[30] Hao Xu, Lei Zhang, Oluwakayode Onireti, et al. 2020. BeepTrace: Blockchain-enabled privacy-preserving contact tracing for COVID-19 pandemic and beyond. *IEEE Internet of Things Journal* (2020).

[31] Ying Yan, Changzheng Wei, Xuepeng Guo, et al. 2020. Confidentiality Support over Financial Grade Consortium Blockchain. In *Proc. of ACM SIGMOD*.

[32] Ce Zhang, Cheng Xu, Haixin Wang, et al. 2021. Authenticated Keyword Search in Scalable Hybrid-Storage Blockchains. In *Proc. of IEEE ICDE*.