A Blockchain Based Architecture for IoT Data Sharing Systems

Mingyu Hou*, Tianyu Kang*, and Li Guo*

*Beijing University of Posts and Telecommunications
Key Lab of Universal Wireless Communications, Ministry of Education
houmy2017, kangtianyulm, guoli@bupt.edu.cn
Li Guo is corresponding author

***Abstract*—Blockchain as an emerging distributed protocol has been widely used in IoT systems. Using blockchain as an IoT data sharing protocol provides precious features including consistency, reliability, and traceability. However, such combination brings high cryptography overhead and consensus latency when sharing data from large number of IoT sensors. To address these issues, we propose a novel blockchain based architecture for IoT data sharing systems. In this architecture, data messages signed by IoT sensors are packaged into data blocks and distributed to the blockchain network. We propose a data block structure with identity-based aggregate signature to protect data reliability from malicious sink nodes and reduce the communication, storage, and computing cost of signatures. We also present a multiple state chain structure with a new consensus algorithm which cuts back consensus phases and accelerate the consensus process. Finally, we evaluate the proportion of data in a block and the blockchain consensus latency, which shows a better performance than PBFT in this scenario.**

***Index Terms*—blockchain, IoT, data, sensor, consensus, aggregate**

I. Introduction

In recent years, the Internet of Things is a rapid developing technology and has played an important role in many fields such as smart industry [1], smart life, and smart city [2]. At the same time, as an emerging technology, blockchain is developing rapidly and applied in many fields. The inherently distributed nature of blockchain and Internet of Things makes them to combine effectively. The introduction of blockchain technology can bring more decentralization, reliability, and consistency especially to IoT data sharing systems. Data reliability and consistency are important requirements in a large number of IoT applications. For example, in the application of food information tracing system, data reliability is even more crucial [3]. Violation of food production regulations or incorrect records of food information will directly threaten the health of consumers. However, it is difficult to construct a unified enterprise or organization to manage between the producers, transporters and sellers in the entire supply chain. In the process of data collection and distribution through the Internet of Things network, data reliability is not ensured. There are major security threats, such as malicious participants tampering with data and attacks on the data center causing a single point failure. Combining with blockchain, the data reliability and consistency are improved with cryptographic algorithms, blockchain data structures and the consensus process.

There are already examples of research and practice as follows. Users can interact with industry IoT devices through a blockchain system to control and monitor [4]. In [5], an energy blockchain system is designed for IoT P2P energy trading. IoT data can also be recorded on cloud storage and blockchain is used for access control and permission management [6].

With limited computing, network, and storage resource, IoT sensors are not able to act as blockchain nodes. In most existing researches in blockchain based IoT systems, sensors transmit transactions through sink nodes to the blockchain network [7], [8], [9]. However, there are still issues combining the Internet of Things with blockchain with this approach.

Firstly, the sink nodes cannot be completely trusted. Data messages are signed by IoT sensors to ensure data reliability against malicious relays. Size of each signature is at least larger than the cryptography hash of the original message. When the number of accessing sensors grows, digital signatures bring a lot more communication and storage overhead to the blockchain network. In our proposed data sharing architecture, IoT devices submit data messages with their signatures, and the sink nodes compress signatures from different sensors by using an identity-based aggregate signature scheme, and package data into blocks for the consensus process. As only a single aggregate signature is required to ensure security of multi-messages, the storage complexity of aggregate signature is $O(1)$, while ECDSA [10] used in Bitcoin [11] and Ethereum [12] has a storage complexity of $O(n)$ which takes a large portion of a block size. Instead of containing all signatures in the blockchain, the storage, communication, and computing costs to ensure data reliability are reduced.

Another issue is that some real-time IoT data is required to be distributed rapidly among system participants. However, some widely used blockchain consensus algorithms like PoW [13], DPoS, and PBFT [14] cannot meet some specific fast distribution requirements. In our considered system, an IoT sensor with limited network condition is only able to connect to a single blockchain node server. Under this scenario, we reconstruct the blockchain structure and propose a new consensus algorithm with lower latency and higher throughput. Our proposed architecture restructure the blockchain public ledger

and is able to tolerate Byzantine faults [15] of the blockchain nodes.

The contributions of this paper can be summarized as follows:

- We propose a new block structure and construction method. An identity-based aggregate signature scheme is used to package messages from IoT sensors to reduce the size of signatures and the verification complexity.
- We propose a new blockchain structure with multiple state chains and design a consensus algorithm, that cuts back phases of the consensus process and reduce consensus latency.
- We implement an experiment blockchain network to analyze and evaluate the system performance.

The remainder of this paper is organized as follows. Section II provides an introduction to blockchain protocols and the cryptography schemes applied in our system. Overview of the data sharing system and major threats to data reliability and consistency are provided in Section III. Section IV presents the baseline protocols of the system and Section V evaluates its performance. Section VI concludes this paper.

II. Preliminaries

A. Blockchain Protocol

Blockchain has a distributed data structure, that multiple nodes store data in the form of a public ledger. Public ledger has a chain structure where transactions were packaged into a block which refers to the former block's cryptography hash. Any update of the public ledger requires a new block proposed and a consensus process for the proposal. The consensus block will then be recorded to the public ledger. Blockchain protocols can be divided into permissioned blockchain protocols and permission-less blockchain protocols according to whether members need to be authenticated. Any person or organization is able to join or quit a permission-less Blockchain (e.g. Bitcoin and Ethereum). While in permissioned blockchains (e.g. Hyperledger Fabric [16], Tendermint [17]), only authenticated nodes are able to interact with the blockchain network. This is generally guaranteed by a public key infrastructure. In our considered scenario, the number of organizations associated with certain IoT data is limited and they have to be recognized in advance, so the permissioned blockchain protocol is more suitable for this scenario. In permissioned blockchain protocols, the consensus algorithm is a form of state machine replication, which maintains the consistency and correctness of every node's local ledger against faults and malicious attacks. It has been proved that the consensus algorithms tolerating Byzantine faults in an asynchronous network condition (e.g. PBFT, Honeybadger [18], Hotstuff [19]) is able to keep safety and liveness with no more than f faulty members out of $3f+1$ member in total [20].

B. Aggregate Signature

Aggregate signature [21] is a digital signature scheme that compresses different signatures into a single signature. The original signatures can be signed by different users on multiple messages. Correctness of each message can be confirmed by verifying a single aggregate signature. In flexible aggregation schemes [21] based on bilinear maps, the aggregating process takes multiple original signatures and anyone can compact them in any order into an aggregated signature.

C. Identity-Based Cryptography

Identity-Based Cryptography (IBC) [22] is a class of public key and certification schemes, that every user takes an unique string like an ID as his public key. IBC needs a Private Key Generator (PKG) to initialize the "master private key" and generate every user's private key with the "master private key" and their IDs. PKG has to be a trusted third party to keep the "master private key" a secret to maintain safety. In an identity-based signature scheme, messages are signed by user's private key, and the signatures are verified with PKG's public key and the signer's ID. As a result, a verifier do not have to check public keys and certificates of all associated signers.

III. System Overview

We consider a scenario where multiple participants share and store IoT data. IoT data is generated by sensor nodes and shared by the participants maintaining the sensor nodes to the other participants that use or monitor these data.

The system network structure is shown in Fig. 1. The sensor nodes are wired or wirelessly connected to sink nodes, which are equipped with or connected to servers. These IoT sensors nodes, sink nodes and servers are maintained by some participants called data producers. For example, in a product information sharing system, product factories managing IoT sensors are the data producers. An IoT sensor is only able to connect to a single data producer because of limited network condition. The blockchain network consists of servers of every system participant called *blockchain node*. Considering the instability of the network environment between servers, the network communication between these blockchain nodes is point to point with a partially synchronous model, where fixed upper bounds Δ on the time to transmit messages between nodes exist but are not known a priori [23].

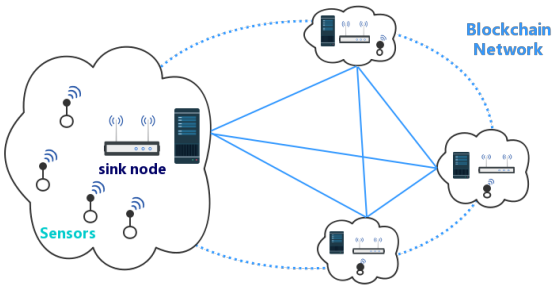

Fig. 1. System network structure.

For data structure, the public ledger of the blockchain system is divided into several *state chains*, as shown in Fig. 2. A state chain shares data from IoT sensors maintained by a single

data producer. All state chains derive different blocks from the same genesis block and add up to contain data from all sensors in the system. On each state chain, the data producer acts as a $proposal\ node$, and nodes of the other participants in the system act as $endorsement\ nodes$. The proposal node collects and packages the sensor messages into blocks and distributes them to all endorsement nodes through the consensus process. Endorsement nodes participate in consensus and verification, and can selectively save all blocks or just block abstracts called $headers$. In the data sharing system, there are multiple state chains parallelly running to share data from different data producers. A data producer playing the role of a proposal node on one state chain is an endorsement node on other state chains.

An IoT sensor with limited communication ability is only able to connect to a single data producer. So only this data producer has the authority to update data on its state chain. Other endorsement nodes cannot directly receive messages from the corresponding sensors, but are able to check and record the consensus data blocks. When the data producer disconnects, the endorsement nodes of the state chain wait until the data producer is restored. During this period, other state chains in the system are not affected. Under this structure, on a certain state chain, there is only one node proposing blocks. The extra complexity caused by changes of the proposal node in general BFT consensus algorithms is avoided. Specifically, our consensus algorithm called SCBFT is described in Section IV.

The blockchain structure and consensus process are applied to maintain data consistency and reliability among all participants. In a data sharing system, there may be a collaborative or regulatory relationship between various endorsement nodes. Data consistency requires that the IoT data is identically recorded on all non-faulty nodes, and is correctly stored and cannot be tampered with. Under our consideration, data consistency and correctness are mainly threatened by malicious proposal nodes modifying data received from sensors or sending different blocks to endorsement nodes. In addition, the endorsement nodes may also be faulty or cooperate with a malicious proposal node due to interest factors. Our proposed architecture ensures the consistency and correctness of IoT data under such threats.

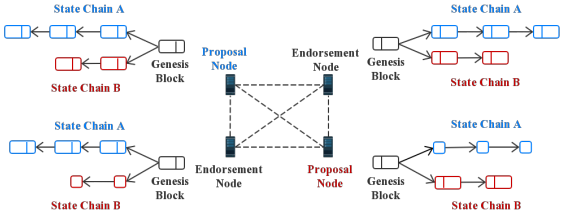

Fig. 2. State chain structure.

IV. Baseline Protocols

A. Data Block Package

We consider that a certain data producer maintains s IoT sensors, indexed by $i \in [s]$, where $[s] = \{1, ...s\}$. In order to deal with the scenario of a large number of sensor nodes access, we propose a scheme for packaging sensor data and constructing data blocks. We apply aggregate signature scheme to ensure data reliability and enable blockchain nodes to verify data messages. Each data message from IoT sensor contains a sensor signature and signatures are compressed during the data packaging process. Aggregate signature scheme reduces the signature storage cost, but still exist a problem. In general PKIs, sensor public key is not directly associated with sensor identity, but a random string. As a result, there needs to be a trusted certificate authority. Sensor certifications are generated to prove the correspondence between the public key and its identity. With a large number of sensors in the system, nodes have to take too much communication, computing, and storage cost to maintain certifications. In our proposed system, a flexible identity-based aggregate signature scheme [24] is applied to alleviate this problem. The unique sensor ID string is the sensor public key and certifications are not required to verify sensor identity.

Based on the identity-based aggregate signature scheme, we propose a data block structure and a block packaging scheme. Before the data sharing system runs, it requires a cryptographic initialization process. During the process, a PKG is formed and generates the master private key and its public key. Then PKG generates and distributes every sensor's private keys to the sensor devices. Private keys of the IoT sensors are encrypted during distribution and storage to ensure that data producers cannot obtain the private keys. It prevents data producers from counterfeiting IoT sensor signatures and tampering with IoT data. PKG's public key is also distributed to every blockchain node.

After the initialization, IoT sensors are able to periodically submit data by sending messages to sink nodes. The message has the form $msg < ID,\ data,\ sig >$, where $data$ indicates recent IoT data, ID is an unique string of a certain sensor, sig is a pairing based signature. Blocks constructed with sensor messages are shown as Fig. 3. A block consists of a header and a body. The block header is an abstract of the block, that contains the hash of the previous block called $prehash$, the hash of the root of the Merkle tree [25] called $root$, a $timestamp$, the data producer's signature for $root$ called $rootsig$, and the aggregate signature called $aggsig$. The block body is a set of transactions $TX < ID,\ data >$, based on $\{msg_i\}_{i\in[s]}$ without the sensor signatures. The $prehash$ connects blocks to its former block, which marks the sequence of blocks in the state chain. The $merkleroot$ and the $rootsig$ prevent transactions from being tampered with. The Merkle tree is constructed based on all transactions in a block. Any change of a transaction will cause the corresponding $root$ changed and cannot match the $rootsig$. The aggregate signature is generated

with messages of IoT sensors, and the PKG's public key based on the identity-based aggregate signature scheme.

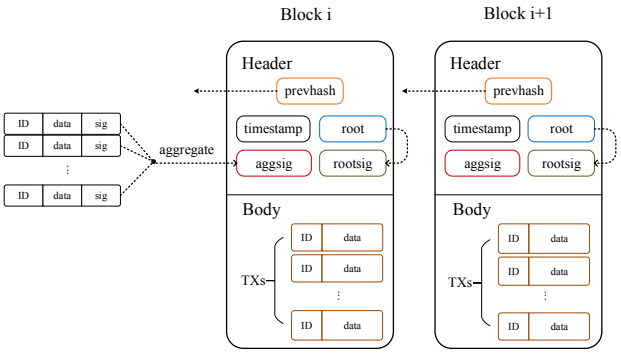

Fig. 3. Block structure.

B. SCBFT Consensus Algorithm

On each state chain, we propose a SCBFT consensus algorithm to keep data reliability and consistency. We consider a general consensus model consisting of a fixed set of $n = 3f+1$ nodes, indexed by $j \in [n]$ where $[n] = \{1, ...n\}$. A set $F \subset [n]$ of up to $f = |F|$ nodes are Byzantine faulty, and the remaining nodes are correct. SCBFT consensus process consists of two phases, a $presynchronize$ phase and a $synchronize$ phase. These two phases ensure that blocks distributed to different endorsement nodes are identical and are able to pass the verification conditions.

a) Normal Case Operation: The working flow of the consensus process in normal case is shown as Fig. 4. During the presynchronize phase, the proposal node broadcasts a proposal block to every endorsement node. Endorsement nodes receive and verify the proposal block according to the top block of its local state chain. Verification mainly includes three parts as follows:

- The $prehash$ equals the hash of the top block on local state chain.
- The $aggsig$ passes the identity-based aggregate signature verification.
- The $merkleroot$ is correctly computed and the $rootsig$ passes signature verification.

A proposal block passing the verification is presynchronized on this endorsement node. The endorsement node record the presynchronized block to its local log and broadcast a presynchronized message to all the other endorsement nodes. It finishes the presynchronize phase and keeps listening to other endorsement nodes.

In the synchronize phase, the endorsement nodes collect presynchronized messages sent from other nodes. Presynchronized messages from different nodes are recorded to log. The proposal block is synchronized if there are more than 2f presynchronized messages for the same block in log. Finally the proposal block is added to the previous block on its local state chain. An endorsement node is able to set a single block synchronized, because there won't be different blocks obtaining enough presynchronized messages in a single consensus process on a state chain. The thredshold of $2f$ presynchronized messages in log guarantees that at least $f+1$ correctly working nodes presynchronize for this proposal block. So if any other proposal block is also synchronized, there should be another $f + 1$ correct nodes presynchronized for this block. A non-faulty node will not be presynchronized for different blocks and send presynchronized messages for different proposals. As a result, if two different blocks are synchronized, there should be more than $3f + 1$ presynchronized messages, which is not possible with $n = 3f+1$ nodes in total in the state chain network. Besides, the 2f threshold also guarantees that an identical proposal block is synchronized on correct endorsement nodes. If correctly working nodes synchronize on different blocks, it also violates the network condition in a similar way as mentioned above. As a result, the consensus process is able to maintain block consistency among correct nodes when there is no more than f faulty nodes in the network. At the end of the synchronize phase, endorsement nodes reply to the proposal node and finish the consensus process, and the proposal node accomplish this proposal after receiving more than $f + 1$ replies and carry on the next proposal. Collecting $f + 1$ replies ensures that the proposal block is synchronized on at least one correct node, which implies that all properly working nodes have reached consensus on the same proposal block.

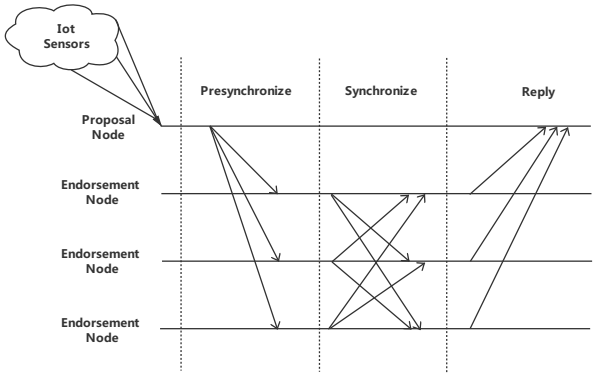

Fig. 4. Normal case operation.

b) Reconnection: SCBFT is able to endure f Byzantine faulty nodes, which implies faulty nodes may disconnect to other nodes or crash due to system errors. An endorsement node may disconnect, and during this period, the state chain reaches consensus on a new block. When this endorsement node reconnects, it will receive a proposal block, whose $prehash$ does not match the endorsement node's top block on its local state chain for missing some blocks during the disconnection. Under this circumstances, the reconnected endorsement node ask other endorsement nodes for those missed blocks and rejoin the consensus process. When the endorsement node successfully adds a new block to its local state chain after a regular consensus process, it checks the

hashes of those blocks received from other endorsement nodes. If $prehash$ of the new consensus block equals the hash of the former block, the endorsement node accomplishes the reconnection process to this state chain. If any modified block was received from a malicious endorsement node, hashes of the blocks won't match with $prehash$ of the consensus block. Then the reconnected endorsement node has to request the block from some other nodes in the system.

The proposal node may also suffer network disconnection. After reconnecting to the network, the proposal node continues its consensus process according to its local log. During disconnection, the proposal node may have missed the relpies from the endorsement nodes. As a result, the proposal node may repropose an already consensus block and is able to receive enough replies and move on to a new block.

V. Evaluation

A. Data Proportion

We evaluate that in our designed data structure sensor data takes a higher proportion of blocks in state chains. In most blockchain systems like Bitcoin, Ethereum, and Hyperledger Fabric, every transaction contains at least one digital signature to certificate its correctness and authorization. While in our system, each block contains a single aggregate signature that maintains the data reliability. We compared the IoT sensor data proportion in a block with ECDSA using the same elliptic curve. IoT data size in each transaction is 64 Bytes, and the results are plotted in Fig. 5. The aggregate signature scheme can significantly reduce the size of signatures in a block and raise the data proportion. The data proportion goes higher when there is larger amount of transactions, and becomes more than 90% with more than 100 transactions in a block.

Fig. 5. Sensor data proportion in a block with or without aggregate signature.

B. Blockchain Network

We analyze the performance comparison between SCBFT and PBFT. Both algorithms in normal case operations have a communication complexity of $O(n^2)$. We measure time to consensus on a single state chain based on these two algorithms with different number of transactions. In the experiment network, blockchain nodes are running on virtual machines and the point to point communication delay is 20 ms. The result is plotted in Fig. 6. It shows that the consensus latency based on SCBFT is lower than PBFT. Because PBFT needs two rounds of message broadcasting and receiving among endorsement nodes, but SCBFT requires only one round. As shown in Fig. 6, it takes longer time to consensus when there are more transactions in a block. Because during consensus, the signature verification and hash verification complexity is linear to the number of transactions.

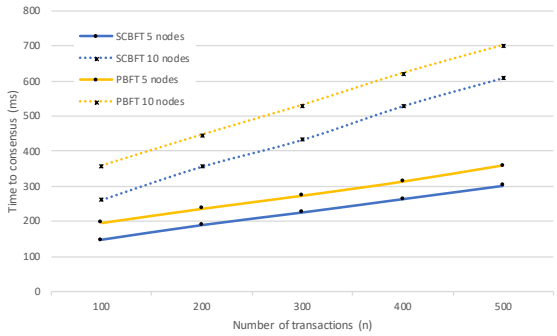

Fig. 6. Time for a block consensus based on PBFT and SCBFT on a single state chain.

In our proposed blockchain network structure, multiple state chains based on SCBFT are able to running in parallel. We also compare this with a single primary node PBFT structure. Our experiment network consists of 5 blockchain nodes and 3 of them are data producers trying to distribute data transactions at the same time. The experiment network condition is the same as above, and the consensus time results is plotted in Fig. 7. The result shows that considering the communication

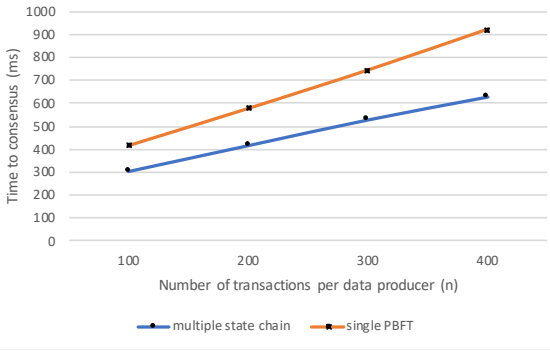

Fig. 7. Time for a block consensus based on multiple state chain structure and single PBFT structure.

and computing cost, our proposed multiple parallel state chains network still has a better performance in consensus speed in normal case operations.

Moreover, the proposal node replacement protocol in PBFT incurs $O(n^3)$ communication complexity and the whole system can not propose new blocks during this period. In our presented blockchain architecture, a faulty proposal node can only affects one corresponding state chain. Sensors maintained by other data producers can still submit data to their chains without waiting for the faulty node to recover.

VI. Conclusion

This study developed a new architecture to share reliable IoT data among system participants based on blockchain. In

this architecture, we use identity-based aggregate signature scheme to ensure data reliability against malicious sink nodes. It also reduces signatures in blocks to a fixed size, which is especially important when collecting data from a large amount of IoT sensors. We also separate the public ledger into different state chains and propose a new consensus algorithm to reduce the communication complexity and the latency of the consensus process. General consensus algorithm like PBFT requires three phases consensus process and two rounds of endorsement nodes broadcasts process, while we reduce the consensus phase to two and the broadcast round to one. Overall, the proposed architecture ensure IoT data consistency and reliability with the blockchain protocol and reduces the computation, communication, and storage overhead of the blockchain based data sharing system.

ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China (No. 2019YFB1406500), Beijing Natural Science Foundation (No. 19L2032), State Scholarship Fund (No. 201906475006) and Shandong Key Research and Development Program (No. 2019JZZY020901).

REFERENCES

[1] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[2] H Arasteh, V Hosseinnezhad, V Loia, A Tommasetti, O Troisi, M Shafie-Khah, and P Siano. Iot-based smart cities: a survey. In *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, pages 1–6. IEEE, 2016.

[3] Zhibo Pang, Jun Chen, Zhi Zhang, Qiang Chen, and Lirong Zheng. Global fresh food tracking service enabled by wide area wireless sensor network. In *2010 IEEE Sensors Applications Symposium (SAS)*, pages 6–9. IEEE, 2010.

[4] Arshdeep Bahga and Vijay K Madisetti. Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications*, 9(10):533, 2016.

[5] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE Transactions on Industrial Informatics*, 14(8):3690–3700, Aug 2018.

[6] Hossein Shafagh, Lukas Burkhalter, Anwar Hithnawi, and Simon Duquennoy. Towards blockchain-based auditable storage and sharing of iot data. In *Proceedings of the 2017 on Cloud Computing Security Workshop*, pages 45–50. ACM, 2017.

[7] Pietro Danzi, Anders Ellersgaard Kalor, Cedomir Stefanovic, and Petar Popovski. Analysis of the communication traffic for blockchain synchronization of iot devices. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[8] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. Blockchain for iot security and privacy: The case study of a smart home. In *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, pages 618–623. IEEE, 2017.

[9] Pietro Danzi, Anders E Kalør, Čedomir Stefanović, and Petar Popovski. Delay and communication tradeoffs for blockchain systems with lightweight iot clients. *IEEE Internet of Things Journal*, 6(2):2354–2365, 2019.

[10] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, 1(1):36–63, 2001.

[11] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.

[12] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[13] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 3–16. ACM, 2016.

[14] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[15] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[16] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30. ACM, 2018.

[17] Jae Kwon. Tendermint: Consensus without mining. *Draft v. 0.6, fall*, 1:11, 2014.

[18] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of bft protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 31–42. ACM, 2016.

[19] Maofan Yin, Dahlia Malkhi, Michael K Reiter, Guy Golan Gueta, and Ittai Abraham. Hotstuff: Bft consensus in the lens of blockchain. *arXiv preprint arXiv:1803.05069*, 2018.

[20] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. Technical report, Massachusetts Inst of Tech Cambridge lab for Computer Science, 1982.

[21] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 416–432. Springer, 2003.

[22] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques*, pages 47–53. Springer, 1984.

[23] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)*, 35(2):288–323, 1988.

[24] Craig Gentry and Zulfikar Ramzan. Identity-based aggregate signatures. In *International workshop on public key cryptography*, pages 257–273. Springer, 2006.

[25] Ralph C Merkle. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*, pages 369–378. Springer, 1987.