Networks of Trusted Execution Environments for Data Protection in Cooperative Vehicular Systems

Philippe Boos and Marc Lacoste

Orange Labs, 92326, Chatillon Cedex, France
{philippe.boos, marc.lacoste}@orange.com

Abstract. Networks of autonomous vehicles roaming in smart cities raise new challenges for end-to-end protection of data in terms of integrity, privacy, efficiency, and scalability. This paper provides a survey of Networks of Trusted Execution Environments (NTEE) architectures. NTEE combine the strong, hardware-rooted security guarantees of the TEE deployed locally in the vehicle, with the distributed protection of a decentralized consensus protocol. We identify three main families of consensus protocols and analyze their architectures, performance, and security, including improvements brought by the TEE. Overall, voting protocols tend to be more efficient for smaller networks, while lottery-based schemes are not easy to apply in a vehicular context due to higher overheads. Both types of protocols reach an intermediate level of security, with variations in byzantine tolerance and types of threats. Graph-based protocols tend to achieve both efficiency and flexibility in terms of network topology support, but their security still remains to be explored.

Keywords: Vehicular networks, data protection, consensus protocols, trusted execution

1 Introduction

Networks of autonomous vehicles roaming in smart cities raise new challenges for end-to-end protection of data. Such hyper-connected, decentralized systems produce and collect from the environment an increasing amount of data, analyzed, stored, and shared with the various stakeholders of the vehicular ecosystem. Examples include itinerary, speed, passenger personal data, network state, or road traffic conditions. Such data is unfortunately highly vulnerable. Protection should strike the right balance between data integrity, intimately related to safety to avoid hazards in autonomous decision-making, and privacy to minimize personal data collection [21]. Protection should also be practical and scalable, supporting different network topologies, both centralized and decentralized.

Networks of Trusted Execution Environments (NTEE) security architectures couple isolation mechanisms such as a TEE (e.g., Intel SGX enclaves) deployed in the vehicle, with a distributed coordination protocol (e.g., blockchain) [8, 36].

NTEE provide both a local secure environment with strong security rooted in hardware (e.g., trusted execution, secure storage, attestation) and decentralized protection such as distributed data integrity. However, it remains unclear which class of protocol matches the previous goals and related trade-offs.

A number of comprehensive surveys already addressed some elements of the problem – for instance, on attacks and counter-measures for vehicular networks (e.g., [22, 23]), on consensus protocols, either from a broad perspective [36], including TEEs, or from an IoT standpoint [20], but not specifically for autonomous vehicles, or through more specific studies on blockchain applications for automotive [29]. However, we felt the need for a shorter, more focused, overall view giving a preliminary assessment of NTEE architectures for protecting data in networks of autonomous vehicles, highlighting research trends and challenges from a literature review, to be confirmed by simulation results in a second step.

Several families of well-known approaches also provide elements of solutions, such as for V2X isolation and trust management (e.g., in-vehicle ECUs isolation, trusted computing [15], often TPM-based, remote attestation, HSMs [35] to guarantee strong, certified security profiles for vehicle or roadside units [10, 11]), privacy (e.g., PKIs [16], pseudonym systems [30]), and resilience [2, 5]. Those solutions are already well investigated, but beyond the scope of this paper.

This paper surveys existing NTEE architectures for cooperative autonomous vehicles, combining the fields of TEEs and consensus protocols. We distinguish three main families of protocols, voting-, lottery-, and graph-based. We discuss their architectural, performance, and security properties. We also show how such properties may be improved using the TEE to meet vehicular constraints, e.g., enhancing overall efficiency for distributed computation of trust [8].

2 System Overview and Approach

Fig. 1. NTEE network topologies

NTEE are distributed systems, composed of nodes, each hosting a *TEE*, and coordinated by a *distributed consensus protocol* [7]. The aim is to reach agreement between all participants in the vehicular network. Different network topologies are possible (see Fig. 1). We assume the network provides notably 5G connectivity, including to the infrastructure, e.g., traffic lights, smart city sensors – wireless connectivity is also possible, e.g., for vehicle-to-vehicle interactions.

The NTEE landscape may be captured using the following taxonomy (see Fig. 2). For each class of consensus protocol, we analyze its inherent architecture, performance, and security properties.

Fig. 2. NTEE taxonomy

Architecture We identify three families of protocols sharing similar architectural features. For each family, we sketch the protocol principle, and discuss its scalability, dynamicity, and flexibility to support several network topologies.

Traditionally, the *weak synchrony* assumption is made: messages may be delayed, duplicated, delivered out-of-order, or lost. They could also be forged by byzantine nodes. In Vehicle-to-Everything (V2X) networks, the *intermittent synchrony* seems more relevant: messages are sent on average within bounded time, while allowing this constraint to be relaxed during some short periods [26].

Several *network topologies* should be supported by the NTEE, ranging from centralized (e.g., star networks), decentralized (e.g., mesh networks), to hybrid. Switching topologies may be possible depending on the vehicular environment or traffic:

- In *smart cities*, vehicles may easily connect to the infrastructure and/or network, as coverage is dense in urban areas. Traffic may be managed smoothly mostly in a centralized manner, with also strong locality (e.g., 4G/5G femtocells, edge technologies).
- In *city borders*, vehicles may reach the infrastructure through another vehicle. Both centralized and decentralized topologies are possible.
- In *scarce connectivity environments*, such as rural areas or zones where 5G infrastructure has not yet been deployed, topologies tend to be fully decentralized, as autonomous vehicles must coordinate peer-to-peer.

Performance Key protocol KPIs are *latency*, *bandwidth*, and *energy efficiency*. *Latency* is the end-to-end time to reach agreement. Low-latency is critical for vehicular cooperative decision-making at high speed, with direct impact on passenger lives, such as in collision avoidance settings (e.g., evasive maneuvers, emergency braking). *Bandwidth* efficiency is also required: according to use cases, data sizes can be huge (e.g., high-resolution maps). *Energy* is consumed by intensive computational tasks (e.g., cryptographic operations, proof-of-work hashing) and network use (with variations between wired/wireless connectivity).

Security and privacy *Confidentiality* is ensured by how nodes access the distributed ledger, either permissioned or permissionless. This property is related to the network architecture, and to how easily new participants may be added in the consensus protocol.

Confidentiality may be guaranteed through *encryption*, before uploading data to the distributed ledger; or through *isolation* by sharing data on a private ledger only between a selected subset of participants. Encryption also allows sharing private data using public key cryptography, but may not be applicable to some IoT devices due to the induced overhead. Current ITS PKI systems also issue multiple public keys for the same ITS station, which could as well severely limit performance. For isolation, in some cases, nodes not part of a private group may be allowed to take part in the computation to reach agreement.

Distributed *integrity* of data manipulated by NTEE nodes is also needed. Blockchain protocols have notably been used for providing vehicular data immutability [20]. Reputation mechanisms within vehicles are also useful for event linkability and anonymity [34].

Availability is essential for reliable cooperation between vehicles, e.g., for exchange of information and decision-making. Fallback mechanisms are needed when the vehicle cannot reach the network nor the infrastructure [2], due for instance to communication or back-end systems failure, in local MEC or cloud servers. Solutions include bounded-time recovery [13] or self-stabilization [14] to guarantee the network of vehicles stays in a safe state. Failures due to interpretation by the vehicle of its rapidly changing environment, added to internal software and hardware flaws, frequent real-time updates, and complexity of multiple autonomic loop orchestration are still major challenges ahead. The platooning case has been particularly investigated in terms of solutions [5].

Privacy is a key challenge in vehicular networks, guided by a principle of data minimization. The aim is to regain control over data collected, shared, and used in the large, multi-stakeholder V2X ecosystem, to protect vehicle identities, avoid vehicle tracking, and preserve driver and passenger attributes against unauthorized nodes [21]. The full spectrum of Privacy-Enhancing Technologies (PETs) is applicable as counter-measures. Homomorphic encryption, secure, verifiable, privacy-preserving multi-party computation (e.g., for committee-based proof-of-stake [36]), differential privacy, hardware protection mechanisms, partial observability, pseudonymity and anonymity techniques are just but a few. While currently deployed V2X solutions mostly rely on the use of pseudonyms [3, 30], a number of consensus protocols are essentially based on some form of leader election mechanism. This might be a challenge for vehicular networks, where nodes constantly change identities. A secure multi-party computation scheme could help to implement the election algorithm in a privacy-preserving manner. Promising solutions for the platooning case have notably been explored [31].

Sybil attacks could be used to hijack such protocols based on cooperation: an attacker could generate any number of ghost vehicles in its neighborhood to vote for him. Available counter-measures are either based on resource testing, location or position verification, or encryption and authentication [18].

3 Voting-Based Consensus

Architecture Participants *vote* to elect a leader in charge of executing a command on the distributed ledger. *Practical Byzantine Fault Tolerance (PBFT)* is the first voting consensus working with weak synchrony assumptions [12]. Since then, many variants have been proposed [24, 26, 36, 37]. To reach agreement, PBFT is based on three rounds of message exchanges (pre-prepare, prepare, and commit). This guarantees that commands are atomically executed and strictly ordered, resulting in a final-state consensus.

Voting-based consensus protocols are generally considered to have *limited scalability* in terms of number of nodes [17]. Scalability was also not much explored beyond $n = 10$ to $n = 20$ nodes [33]. Informally, every participant joining the network has to be acknowledged by all others. Adding (or removing) nodes scales as $O(n^3)$, due to such intensive all-to-all network communications. Such protocols could therefore be very expensive for V2X where nodes are highly mobile, continuously switching from one area to another. Scalability may be greatly improved with optimizations such as canonization of phases enabling pipelining, communication complexity in each phase being reduced from $O(n^2)$ to $O(n)$ [37].

In terms of *topology flexibility*, PBFT schemes are expected to be efficient in environments with scarce network coverage: agreement is reached rapidly when the number of participants is kept small. In an urban environment, they could be applicable to manage traffic at intersections (e.g., smart traffic lights) to take fast decisions. They seem less relevant in smart cities when there are many vehicles to coordinate.

Performance *Latency* depends on the time to perform a commit on the ledger. Thus, as for *bandwidth*, it is efficient while the network size is kept small. Due to the $O(n^2)$ message complexity, such protocols are not applicable for much larger networks. Voting-based schemes achieve medium *energy efficiency*: while most of the energy costs are spent in communication, many messages still have to be exchanged to reach agreement.

Security Such protocols are immune against *downgrade or rollback attacks* [9]: voting consensus are final-state. Thus, any change committed on the distributed ledger is definitive.

Regarding *byzantine tolerance*, PBFT protocols are proven to tolerate up to 1/3 of byzantine nodes [33]. The main security threat is *DDoS* against the leader. The root cause is the protocol design itself, which is at some points centralized around the leader. The approach of choosing a new leader after a timeout expiration is only crash-tolerant, and not byzantine-tolerant, as the DDoS attack can follow the new leader. Apart from a rapid leader change, there seems to be no evident counter-measure [37].

TEE impact *Security* and *byzantine tolerance* are improved. The TEE shifts trust to the hardware, instead of the decentralized protocol. Enclaves guarantee integrity of the executed code using hardware protection. Mechanisms are also available to attest run-time integrity [1]. Moreover, by running a simple crash-tolerant algorithm (e.g., Paxos, Raft) inside an enclave for every participant, a behavior similar to a byzantine-tolerant algorithm may be achieved [24].

The TEE also improves *performance* by reducing the number of messages needed for agreement. Most TEEs support monotonic counters, theoretically impossible to reset, which guarantee a trusted order for messages: the message number is signed together with the message, allowing honest nodes to sort them, even over unreliable networks. This approach has been explored in systems like FastBFT, where message complexity is reduced to $O(n)$ [24]. It can also be used to improve byzantine tolerance [32].

4 Lottery-based Consensus

Architecture Instead of a vote, the leader is selected by a shared *lottery* algorithm in which all nodes participate. When a solution to the lottery is found by a node, it is sent to all for validity checking. Assuming a majority of honest nodes, an agreement is reached on the data sent by the winning node. The data must be consistent with the ledger current state, or it will be rejected. Each node then adds this data to its local copy of the ledger. Many algorithms are available, known as *proof-of-X*, e.g., *proof-of-work (PoW)*, *proof-of-luck (PoL)* [27], *proof-of-elapsed-time (PoET)* [19]. Hybrid lottery-based solutions may also choose a group of nodes that then elect the leader using a PBFT-like scheme.

Lottery-based protocols are generally more *scalable* than their voting counterparts – despite many variations among the wide range of proof-of-X schemes. For PoW, a new node does not need to register anywhere, and can join the network just by working on the proof. In other schemes (e.g., based on cryptographic sortition [25]), a new node has to broadcast its public key to all others to be part of the selection process, which makes them slightly less scalable.

Lottery-based consensus protocols are also applicable to many *topologies*. They are thus already used for distributed management of vehicular data [25].

Performance Due to *latency*, the applicability of such protocols may be limited for cooperative vehicles: to reach agreement and execute a command on the ledger, the lottery has to be won by a participant. The frequency of winning results should be very high if frequent updates are needed. Nevertheless, PoL and PoET might provide faster agreement between vehicles than PoW. *Bandwidth* efficiency is highly variable, depending on the block generation rate, ranging from weak (PoW mining) to moderate (PoL).

Energy efficiency can be very poor depending on the lottery mechanism. For intensive hashing algorithms (e.g., PoW), there will be a huge waste of energy to reach agreement. With other mechanisms, when the lottery process is based on the participants public keys instead of a looping computation, the energy issue is less significant.

Security A first threat is the occurrence of *forks*: two participants may win the lottery in a short time frame, which may split the ledger into two valid parts when the two results are broadcast simultaneously. Forks may be resolved probabilistically over time, by nodes choosing always the longest chain when the next result is found.

An attacker may also attempt to downgrade the distributed ledger state to a previous one to take control over the latest transactions (*rollback attack*). Mitigation includes using a final state consensus, avoiding transient unsafe states, so that previous states may not be restored. A trade-off must be reached between consistency and partition tolerance, and must be considered in the NTEE deployment policy [17]: sealing blocks will prevent rollbacks, but will also prevent the distributed ledger from merging after a fork. Conversely, to avoid forks, rollbacks must be allowed.

Another threat are *majority attacks*: integrity holds if less than 1/2 of nodes are byzantine. If an adversary takes control of more than 50% of hashing power, consensus may be corrupted by forging blocks with double-spending transactions, or putting transactions in arbitrary orders. There is no real applicable countermeasure over public networks.

TEE impact *Energy efficiency* may be improved by delegating *trust* to a secure enclave rather than to a hash function. Compared to PoW, Intel PoET randomly backs off for an exponentially distributed period of time. The TEE provides hardware attestation that the node has really awaited this time. The PoL approach is similar: all nodes draw a random number in a periodic time slot. Effective randomness is guaranteed by the enclave.

Delegating trust to hardware is also beneficial for *performance*: time- and energy-consuming computing tasks related to proofs-of-X could be highly improved in an enclave, e.g., that may efficiently attest for an amount of work [19].

5 Graph-based Consensus

Architecture New consensus protocols based on *directed acyclic graphs (DAGs)* are emerging. For instance, the *hashgraph* proposes a *virtual voting* consensus [6]. Nodes vote "virtually" for one another, without exchanging specific agreement messages. The DAG structure represents the inner network in memory – vertices being the network nodes, edges the network links, and flows through the graph the unidirectional message communications. The hashgraph advertizes all events in the network. Nodes become interchangeable in terms of knowledge about what happens in the network, hence the virtual voting consensus.

Graph-based consensus protocols appear to achieve the greatest flexibility in terms of support of *network topology*, from centralized to decentralized. Such protocols could therefore be applicable to all network topologies found in vehicular environments.

Performance Virtual voting does not require to exchange messages to reach agreement, which should result in low *latency*. Embedding the consensus metadata in the gossip protocol, responsible for data exchange, is foreseen to achieve very efficient *bandwidth* usage, close to the theoretical limit, i.e., sending only the transaction data [6]. Such protocols should also be very *energy-efficient*, reaping benefits from both voting- and lottery-based approaches: energy is only spent on communication, and network usage is limited thanks to virtual voting. Such a property is particularly interesting for cooperative autonomous systems.

Security Graph-based protocols achieve fair *byzantine tolerance*, being resilient to 1/3 of byzantine nodes and to forks. Another promising property is *resistance to DDoS* due to the fully decentralized architecture. How the TEE could improve performance and security of a DAG-based consensus protocol remains an open area for further research.

6 Conclusion

Fig. 3. Main trends: performance vs. security

Figure 3 shows some broad trends for the consensus families, focusing on the performance vs. security trade-off. Security is captured through *byzantine tolerance*, and resistance to other attacks. In a first step, performance is assessed through *latency*, critical KPI for V2X safety – a finer evaluation, e.g., taking into account bandwidth and energy efficiency is left for future work.

Overall, *voting protocols* tend to be more efficient for smaller networks, while *lottery-based schemes* are not easy to apply in a vehicular context due to higher overheads. Both types of protocols reach an intermediate level of security, with variations in byzantine tolerance and types of threats. TEE usage also improves security and performance. *Graph-based consensus protocols* are promising, notably the virtual voting of hashgraph, as they seem to achieve both efficiency and flexibility, but further investigation is needed on those systems.

For V2X, the three NTEE families are applicable to protect data, depending on network topology, infrastructure and use case requirements. Previous findings will need to be confirmed by simulation results, realistic use cases, or practical deployments for coordination of vehicles. Different classes simulators could be explored, e.g., for vehicular traffic [28], or V2X ETSI ITS-G5 [4]. NS3 is also promising as it allows to clearly separate the consensus layer from other protocol layers thanks to Direct Code Execution.

We expect *flexibility* in the data protection architecture to be needed to support *multiple consensus protocols* – including transparent protocol switching at run-time. Typical examples include: (1) vehicles moving *between environments* (e.g., smart cities, city border, rural area); (2) supporting *other verticals* (e.g., drones, robots, etc.); (3) aiming for different real-world *use cases*. We are thus currently implementing a modular simulation framework supporting multiple protocol families (e.g., PBFT-like, PoL, PoET) and run-time features.

Acknowledgments. We thank Ahmad-Reza Sadeghi and David Koisser for their help and insightful comments on the paper. This work is part of 5GCroCo project funded by the EU H2020 Research and Innovation Programme (grant no. 825050).

References

1. Abera, T., Bahmani, R., Brasser, F., Ibrahim, A., Sadeghi, A., Schunter, M.: DIAT: Data Integrity Attestation for Resilient Collaboration of Autonomous Systems. In: Annual Network and Distributed System Security Symposium (NDSS) (2019)
2. Alam, M., Ferreira, J., Fonseca, J.A.: Intelligent Transportation System (ITS): Dependable Vehicular Communications for Improved Road Safety. Springer (2016)
3. Amro, B.: Protecting Privacy in VANETs Using Mix Zones with Virtual Pseudonym Change. arXiv:1801.10294 (2018)
4. Artery: OMNeT++ V2X simulation framework for ETSI ITS-G5, <https://github.com/riehl/artery>
5. Axelsson, J.: Safety in Vehicle Platooning: A Systematic Literature Review. IEEE Transactions on Intelligent Transportation Systems 18(5), 1033–1045 (2017)
6. Baird, L.: The Swirls Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance. Swirls Tech Report SWIRLDS-TR-2016-01 (2016)
7. Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., Danezis, G.: Consensus in the Age of Blockchains. arXiv:1711.03936 (2017)
8. Brandenburger, M., Cachin, C., Kapitza, R., Sorniotti, A.: Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric. arXiv:1805.08541 (2018)
9. Brandenburger, M., Cachin, C., Lorenz, M., Kapitza, R.: Rollback and Forking Detection for Trusted Execution Environments Using Lightweight Collective Memory. In: IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) (2017)
10. C-Roads: The Platform of Harmonised C-ITS Deployment in Europe: <https://www.c-roads.eu/platform.html>
11. Car2Car Communication Consortium: <https://www.car-2-car.org/>
12. Castro, M., Liskov, B.: Practical Byzantine Fault Tolerance and Proactive Recovery. ACM Transactions on Computer Systems (TOCS) 20(4), 398–461 (2002)
13. Chen, A., Xiao, H., Haeberlen, A., Phan, L.T.X.: Fault Tolerance and the Five-Second Rule. In: Workshop on Hot Topics in Operating Systems (HotOS) (2015)
14. Dolev, S.: Self-Stabilization. MIT Press (2000)
15. E-safety Vehicle Intrusion proTected Applications (EVITA): www.evita-project.org
16. ETSI TS 103 097 V1.3.1: Intelligent Transport Systems (ITS); Security; Security header and certificate formats (2017)
17. Gilbert, S., Lynch, N.: Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. ACM SIGACT News 33(2), 51–59 (2002)

18. Hamdan, S., Hudaib, A., Awajan, A.: Detecting Sybil Attacks in Vehicular Ad Hoc Networks. arXiv:1905.03507 (2019)
19. Intel: PoET 1.0 Specification (2015)
20. Kang, J., Yu, R., Huang, X., Wu, M., Maharjan, S., Xie, S., Zhang, Y.: Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks. *IEEE Internet of Things Journal* 6(3), 4660–4670 (2019)
21. Karnouskos, S., Kerschbaum, F.: Privacy and Integrity Considerations in Hyper-connected Autonomous Vehicles. *Proceedings of the IEEE* 106(1), 160–170 (2018)
22. Kelarestaghi, K.B., Foruhandeh, M., Heaslip, K., Gerdes, R.M.: Survey on Vehicular Ad Hoc Networks and Its Access Technologies Security Vulnerabilities and Countermeasures. arXiv:1903.01541 (2019)
23. Lima, A., Rocha, F., Völpl, M., Esteves-Veríssimo, P.: Towards Safe and Secure Autonomous and Cooperative Vehicle Ecosystems. In: *ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC)* (2016)
24. Liu, J., Li, W., Karame, G.O., Asokan, N.: Scalable Byzantine Consensus via Hardware-Assisted Secret Sharing. *IEEE Transactions on Computers* 68(1), 139–151 (2019)
25. Lundbæk, L.N., Janes Beutel, D., Huth, M., Jackson, S., Kirk, L., Steiner, R.: Proof of Kernel Work: a Democratic Low-Energy Consensus for Distributed Access-Control Protocols. *Royal Society Open Science* 5(8), 180422 (2018)
26. Miller, A., Xia, Y., Croman, K., Shi, E., Song, D.: The Honey Badger of BFT Protocols. In: *ACM Conference on Computer and Communications Security (CCS)* (2016)
27. Milutinovic, M., He, W., Wu, H., Kanwal, M.: Proof of Luck: An Efficient Blockchain Consensus Protocol. In: *ACM Workshop on System Software for Trusted Execution (SysTEX)* (2016)
28. MovSim: <http://www.movsim.org>
29. Ortega, V., Bouchmal, F., Monserrat, J.F.: Trusted 5G Vehicular Networks: Blockchains and Content-Centric Networking. *IEEE Vehicular Technology Magazine* 13(2), 121–127 (2018)
30. Petit, J., Schaub, F., Feiri, M., Kargl, F.: Pseudonym Schemes in Vehicular Networks: A Survey. *IEEE Communications Surveys & Tutorials* 17, 228–255 (2015)
31. Santini, S., Salvi, A., Valente, A.S., Pescapè, A., Segata, M., Cigno, R.L.: Platooning Maneuvers in Vehicular Networks: A Distributed and Consensus-Based Approach. *IEEE Transactions on Intelligent Vehicles* 4(1), 59–72 (2019)
32. Veronese, G.S., Correia, M., Bessani, A.N., Lung, L.C., Verissimo, P.: Efficient Byzantine Fault-Tolerance. *IEEE Transactions on Computers* 62(1), 16–30 (2013)
33. Vukolić, M.: The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In: *International Workshop on Open Problems in Network Security (iNetSec)* (2015)
34. Whitefield, J., Chen, L., Giannetsos, T., Schneider, S., Treharne, H.: Privacy-Enhanced Capabilities for VANETs using Direct Anonymous Attestation. In: *IEEE Vehicular Networking Conference (VNC)* (2017)
35. Wolf, M., Gendrullis, T.: Design, Implementation, and Evaluation of a Vehicular Hardware Security Module. In: *14th International Conference on Information Security and Cryptology (ICISC)* (2011)
36. Xiao, Y., Zhang, N., Lou, W., Hou, Y.T.: A Survey of Distributed Consensus Protocols for Blockchain Networks. arXiv:1904.04098 (2019)
37. Yin, M., Malkhi, D., Reiter, M.K., Gueta, G.G., Abraham, I.: HotStuff: BFT Consensus with Linearity and Responsiveness. In: *ACM Symposium on Principles of Distributed Computing (PODC)* (2019)