Achieving consensus for Liability Attribution on the Blockchain

Ritesh Ahuja and Bhavya Gulati

1. Introduction. Autonomous and driverless navigation technology holds a vast disruptive potential, boasting reduced mobility and infrastructure costs, increased safety, increased mobility, increased customer satisfaction and reduced crime. Vehicles with autonomous driving capability sense their environment and navigate without human input, and as a result generate enormous data from the sensors that identify appropriate navigation paths, as well as obstacles and relevant signage. Techniques used to detect surroundings include radar, laser light, GPS, odometry and computer vision. The output from EDR(Event Data Recorder) and OBD(On board Diagnostics) currently enables car manufacturers to diagnose and improve the accuracy of control systems that utilize the sensor input. There are other means of data inputs also being considered amongst the industry players, some would like self-driving cars to base their decision only on what their sensors signal (as in the case of Waymos purely offline cars, or in line with Apples Autonomous Navigation Systems); some players advocate the use of information received from other cars (Vehicle-to-Vehicle, or V2V communication); and yet another group of players (in particular, European telecommunications companies and the whole of the EU) propose that self driving cars use information received from the infrastructure (Vehicle to Infrastructure, V2I) or from the whole environment (Vehicle-to-Environment, V2E) [14]. The latter implies reliance on 5G networks, LiDAR sensors, and other fixed and wireless infrastructure sources.

One of the critical technological challenges in widespread adoption of driverless technology is the development of trustable dispute and liability resolution in the event of traffic collisions or environmental accidents. The data collected from sensors can be instrumental in detecting the faulty party in post accident scenarios. However, real-world events indicate that this information is almost never shared, even in the event of an accident[20]. In short, there are several entities that could share the blame, and ultimately since an insurance agency needs to payout, they rely on an accurate assessment by the legal authority. However, due to the lack of direct control of the vehicle sensor and manufacturing data, data from witnessing vehicles (V2V) and data from environmental sensors (V2I), all involved entities require trust in the integrity of the information provided by the other party in order to assure an accurate assessment. Recent events have brought these concerns to limelight, wherein due to the lack of trust in the information provided by a potentially liable party, the only resolution is by the way of a lengthy litigation [18]. A compelling alternative is a decentralized system in which participants together ensure integrity by agreeing on the validity of one anothers transactions.

In this paper, we reconcile these challenges by leveraging the blockchain technology to ensure the integrity of forensic data and streamline liability resolution in a trustable manner. Different blockchain protocols employ different methodologies to achieve consensus. In this particular scenario, we propose a transportation consensus protocol that reflects real-world constraints of trust in the automobile industry. The transportation consensus protocol can ensure the integrity of data in the transportation network. We also leverage existing technolo-

gies in cryptographically secure communication to enable private-aware autonomous driving on the blockchain. Lastly, we discuss techniques to reduce data storage cost on the blockchain by utilizing off-chain peer-to-peer distributed file systems.

The rest of the paper is organized as follows. Section 2 lists the involved organizations and the requirements that must be met for trustable liability resolution. Section 3 presents the transportation consensus protocol for AV forensics on the blockchain. Section 4 discusses possible shortcomings of the protocol and proposes some future work. Finally, Section 5 concludes the article.

2. Preliminaries and Related Work. We take into account several entities that interact with each other in this system. The Autonomous Vehicle (AV) provides primary evidence needed for forensic analysis to make liability decisions. The Witnesses (W) are also autonomous vehicles that submit data on events of other vehicles in their surrounding (e.g., the speed of the vehicle which crossed a red light). The Auto manufacturer (AM) receive sensor data information from an AV for diagnostics and to improve the accuracy of autonomous control systems that utilize the sensor input. In turn, it provides over-the-air software updates that improve the driving experience. The Service Center (SC) provides maintenance services for the autonomous vehicle and provides a report as a proof of service to the vehicle. The Insurance Company (IC) pays compensation to the vehicle owner against financial loss in the event of an incident involving a vehicle they own, such as in a traffic collision. The Public Transport Authority (TA) regulates or administers transportation related matters, such as the construction and maintenance smart roadside infrastructures (e.g., stop signs and traffic lights) and cryptographic credentials needed to secure v2v and v2i communication. Finally, the Legal Authority (LA) processes the evidence to make liability decisions in the case of disputes. They make also provide evidence from outside the system (e.g., weather calamities, unforeseen attacks, etc) to facilitate payment of compensation.

Imagine that you are detectives just arrived on site to open a accident scene investigation, and at first glance you see a face-on collision at an intersection of a two-way road on a green light, wherein one of the cars (say AV2) appears to have swerved left into the oncoming car (say AV1) and caused the accident. As investigators, you need to ask yourselves a few questions about the circumstances that led to the incident. For example, what made the collision possible? What did the car know about the conditions involved? And ultimately, who is liable for the damage incurred? The scenario can play out in many ways. Say the driver of AV2 took control of the vehicle and swerved left into AV1, causing the accident; in this case AV2 would be held liable. Alternatively, it could be the case that AV2 swerved left to avoid a third vehicle that drove into the intersection from a red light: this would imply that the third vehicle (AV3) is liable. Perhaps, the third vehicle drove into the intersection because he too saw that the traffic lights were green, in other words a "Green/Green" light intersection accident, clearly an error of the public Transport Authority (TA). In another situation, AV2 ,while autonomously driving, failed to detect the oncoming traffic correctly, in which case the Auto Manufacturer (AM) of the vehicle would be help liable. However, perhaps the manufacturer send out updates to the car to fix these issues, which the driver of the vehicle neglected to apply to his vehicle; in which case AV2 would be help liable. Yet another situation could be that the vehicle underwent some repair at the Service Center (SC)

and now has a left leaning tendency when accelerating, causing the car to swerve left and crash into oncoming vehicle. In which case, the service center should be help liable.

Our crime scene investigation is mostly aimed at determining what happened, more so than who is liable for the damage done. The two results dont come at once: even if the whole accident and causation chain is identified, attributing liability can be a daunting task. Recent work suggests certain principles and guidelines digital forensic must follow[4]. Other work on liability resolution propose utilizing the cloud storage to upload all information related to an event, such that it may simplify dispute resolution[17], however, they do not consider the lack of trust from adversial players in their system and consequently the integrity of the data available. There are two different issues that are worth highlighting in this respect. In some circumstances, a car manufacturer could object that the damage occurred due to an unforeseen event, which even a high-capacity computer could not process; or that external circumstances intervened (e.g. a cyber attack), which did not depend on the negligence of the manufacturer, the vendor, or the transportation company (e.g. if a company like Uber or Lyft had provided the self-driving car) [10]. In this respect, our objective is to 1) ensure integrity of information by making sure transactions are immutable once confirmed, 2) enable non-repudiation such that the parties can be held responsible for their actions by providing proof of the integrity and origin of data, 3) be lightweight for the system to be capable of being secure even with low computational resource. and lastly, 4) be scalable in order to serve hundreds of millions of participants.

However, like all distributed systems, blockchain systems are challenged with network latency, transmission errors, software bugs, security loopholes and black-hat hacker threats. To counter these potential errors, a blockchain system is in need of an efficient consensus mechanism to ensure that every node has a copy of a recognized version of the total ledger. Traditional fault tolerance mechanisms concerning certain problems may not be completely capable of tackling the issue that distributed and blockchain systems are faced with.

The earliest proposal at reaching consensus on a blockchain is the proof-of-work (PoW) scheme, which is used by Bitcoin and Ethereum [19].In PoW schemes the processor must solve a PoW puzzle which has publicly verifiable solution; this interaction forms the core of the security in work-based consensus mechanisms. Because solving PoW requires computation, the number of identities that the malicious processors can create is limited by the fraction of malicious computational power. These are also termed public blockchains as they are completely open and anyone is free to join and participate in the core activities of the blockchain network. Other mechanisms include proof-of-stake (PoS) [15], wherein the participation in the consensus-building process is restricted to parties identified as having a legitimate stake in the blockchain. In the meantime, Delegated Proof of Stake has stepped up as an alternative. DPOS is used in blockchain projects like Bitshares, Lisk, NEO, Ark, Tezos, and EOS. Owners of stakes in a DPOS system take part in a continual voting process to elect a set number of block producers known as witnesses. One of the witnesses is randomly selected by the group or producer to confirm a block per period. Block producers effectively play the role of miners but without all the overheads. However, our system requires high speed and high throughput processing of transactions within a group of known participants, and due to the computation and data storage limitations on the vehicles and the lack of a regulating crypto-currency, PoW, PoS, dPoS and other non-deterministic consensus mechanisms on public blockchains are

unsuitable for our use. Accordingly, we employ "permissioned blockchains", which generally do not involve PoW (that is, blockchains with no mining) or some other system requirement from the nodes.

We utilize existing Public Key Infrastructure such as a certified authority (CA) to issue unique digital identities to all autonomous vehicles to facilitate authenticated and authorized communication. Ethereum and Bitcoins scalability issues stem from their now-inefficient consensus protocols. When the identity of the participants is known, there are efficient and deterministic solutions available such as those adapted in the more recent blockchain systems, e.g., Hyperledger[6], Ripple [5], etc. Distributed fault-tolerant consensus in such a closed settings is a well studied topic in distributed systems. Raft [21], Paxos [16], PBFT [7] are popular protocols that are in active use today, with PBFT (or its delegated variant d-PBFT) being the most popular amongst them. In contrast to the solution in the PoW, PBFT requires all parties on the network (all nodes) to submit their individual conclusions in order for a consensus to be reached. Hence, the classic PBFT protocol is communication bound: $O(N^2)$ where N is the number of nodes, and is known to suffer tremendously when the size of the network grows over 100 nodes[9]. Whereas, we need a consensus protocol that can scale to millions of nodes. Since, the secure nature of blockchain originates from the consensus algorithm employed for appending new blocks into the blockchain, it is the primary the focus of our work.

The on board processors of autonomous vehicles impose a combination of bandwidth and computational limitations. Issues of lightweightedness in terms of storage cost on the blockchain have well established solutions in practice. We discuss them here for the sake of completeness. The bulk of sensor data need not be stored on the blockchain. Instead of the complete transaction data only its hash can be persisted on the blockchain. The hash length is independent of the size of the sensor and environment data submitted. The actual forensic data can be stored on a central server. When accessing the data, their hashes are compared to those on the blockchain. However, this design introduces a single point of failure. A better alternative is to use a a peer-to-peer distributed file system, such as Swarm[11] or InterPlanetary File System [2]. These seek to connect all computing devices with the same system of files. IPFS, for example, provides an immutable, content-addressed block storage model. Each item can be identified by a hash and accessed using an URL. By using a distributed file system, the single authority and point of failure is removed.

Issues of privacy of the blockchain also have relatively well established solutions in the literature [8]. A simple solution is based on IEEE 1609.2[22, 1] whereby, an entity can communicate with other entities without revealing its identity. The IEEE 1609.2 security standard defines a vehicular public key infrastructure (VPKI) for secure vehicular communications to ensure that data exchanges are secured. Data integrity is protected by digital signatures along with a corresponding certificate sent with the data. The certificate is based on pseudonyms to preserve the privacy of the vehicle owner. In practice the validity of a certificate is about 5 minutes thus the verification data sent by an autonomous vehicle would include the validity of the certificate.

In the next section we propose a concensus mechanism especially tailored to the transportation system of autonomous vehicles.

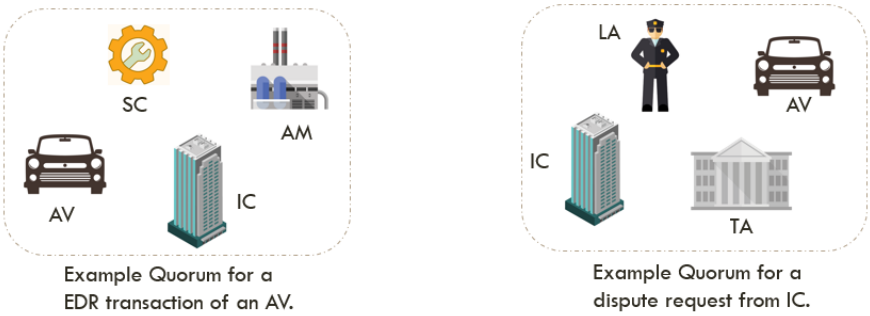

Figure 1. *Examples of quorums such as (left) transaction of data submitted on the blockchain by an Autonomous Vehicle and (right) transaction proposed by an Insurance Company request dispute resolution.*

3. Consensus on the Transportation Blockchain. The Autonomous Vehicles are usually resource constrained, which implies the lack of capability to support computing intensive tasks or keep large stores of data. Hence we propose to use PBFT as a consensus protocol. However, the classic PBFT [7] requires at least a quadratic number of messages in the number of participants, thus they are bandwidth-limited, i.e., more network identities leads to worse performance. Network bandwidth limits the transaction throughputs for a network of even a few hundred nodes severely. Accordingly, recent proposals [8] in digital blockchain forensic that utilize all participants in the network of millions of nodes to arrive at a consensus using PBFT are unscalable in practice. With millions of vehicles and few hundred other establishments composed of legal authorities, insurance agencies, auto manufacturers and service centers who need to reach an agreement on the contents of the blockchain, we need a consensus mechanism wherein the relevant parties in a system. Our consensus protocol emulates real-world relationships of trust in the the automobile industry, while providing security of the blockchain from byzantine behavior.

We utilize the notion of *quorum*. In a distributed system, a quorum is a set of nodes sufficient to reach agreement. For each particular transaction in the system we propose a quorum of participants that need to verify and validate that transaction in order for it to be considered for a block. The intuition is that each participant knows of others it considers important in a liability dispute. Any transaction it proposes must be approved by the vast majority of those others before considering the transaction settled. For example, as driver of an Autonomous vehicle, you rely on the public transport authority to receive traffic communication (e.g., red light signals or obstacle position), you rely on your Auto Manufacturer to receive updates for efficient operation and navigation, you require your Service Center to conduct regular check-ups in order to keep the vehicle running smoothly, and your Insurance Agent to provide financial protection against physical damage or bodily injury resulting from traffic collisions and against liability that could also arise from incidents in a vehicle. And in the case of an accident, these participants form your sphere of influence, and it is only natural for them to first verify your identity and then validate your transactions. Figure 1 illustrates these example quorums, wherein the proposer of the transaction is in the bottom left corner of

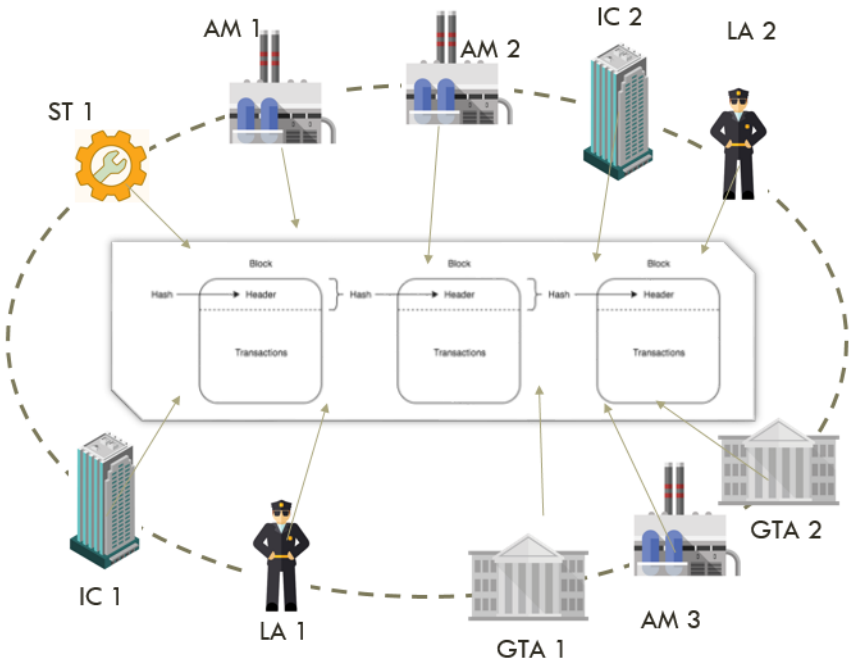

Figure 2. *Full network consensus on the blockchain.*

each quorum. Eventually, every few seconds all transactions from all quorums in the system are brought together into the blockchain by the consensus algorithm (As illustrated in Figure 2. If enough of the network accepts a transaction then it becomes infeasible for an attacker to roll it back. Once consensus is reached, the current blockchain increment by the new block. Assuming that the consensus algorithm is successful, and that there is no fork in the network, the last state of the blockchain maintained by all nodes in the network will be identical.

In particular, protocol proceeds in rounds, wherein in each round a block is committed to the blockchain. In each round:

1. Initially, each entity (except the Autonomous vehicles) brings all approved transactions from all quorums it has participated in. These transactions must have not already been applied (these may include transactions held over from a previous consensus process), and are made public in the form of a list called "quorum set".
2. Each entity then consolidates all quorum sets that are public, and votes on the veracity of all transactions. Note that the transactions have already been validated within their respective quorums but still need to be verified by all other entities before being included in the blockchain.
3. Transactions that receive more than a minimum percentage of yes votes are passed on to the next round, if there is one, while transactions that do not receive enough votes will either be discarded, or included in the quorum set for the beginning of the consensus process on the next block.

4. Discussion. Fault Tolerance. The strength of a consensus algorithm is usually measured in terms of the fraction of faulty processes it can tolerate. It is provable that the PBFT protocol (which already assumes synchronicity, and known participants) can tolerate more than $(n-1)/3$ byzantine faults, or 33% of the network acting maliciously. However, due to the multi-stage construction of our consensus protocol, we are not sure if the fault tolerance of the classic protocol extends to our design.

Protection against evidence tampering. The time between when a transaction is proposed and when it is validated leaves a window of opportunity for the transaction proposer to tamper his data in case he is the likely liable entity. Our proposed consensus protocol addresses this attack by validating transactions as soon as they are successfully verified within the quorum. After the transaction is validated it becomes difficult for the proposer to tamper with validated transactions, while evading detection, before the transaction are confirmed in a block. Nevertheless, in its current state of analysis we can not claim that our protocol is provably secure in the face of byzantine faults. It definitely needs to be vetted through development in an adversarial cryptographic environment.

Adversarial Attacks. Suppose, in the event of an accident such as a multiple collision scenario, an auto manufacturer could after receiving primary evidence collude with its autonomous vehicle by propagating misleading information by altering the contents of host vehicle data to evade liability. This and other similar forms of collusion can be resolved by verifying data sent by a rogue auto manufacturer against the data sent by an insurance company. This is done by computing the hash of data, from insurance company and auto manufacturer and checking the location and timestamp on both data. The outcome will reveal the malicious intention of the rogue auto manufacturer as the computed hash as well as the time stamp will differ. Other forms of verification involves comparatively verifying data with other witnesses involved in the accident i.e. correlating data of an auto manufacturer and insurance company of an involved vehicle with other involved vehicle's auto manufacturer and insurance companies in both time and space by checking timestamps and location on each data. While we discussed just a single instance of abuse, there can exist several adversarial attack vectors that have not been considered due to the limited scope of this project.

Scalability. Indeed, one of the biggest challenges in the integration of blockchain into this application setting is scalability. In fact, due to the massive number of vehicles and resource constraints, deploying blockchain in this environement is particularly challenging. A viable blockchain architecture has to accommodate the many vehicles (they become the peers on the blockchain network), and it should be able to process a high throughput of transactions.

Scalability (in the number of participating nodes). We brought down the number of nodes needed to achieve consensus from hundreds of millions of vehicles and entities to just a few hundred key entities (such as the couple of auto manufacturers and insurance agencies across the country, a few legal authorities that represent each state, couple hundred service centers, etc). However executing PBFT based voting mechanisms might still not scale to meet industrial standards. In these situations a Delegated style BFT protocol has been explored both in practice [3] and in literature [13].

Scalability (in the number of transactions). Due to the real-world nature of traffic and traffic events, transaction generation is infrequent and bursty. One answer to blockchain scaling is sharding. Sharding promises to increase the throughput by changing the way blocks

get validated by the network. The key feature of sharding that makes it unique among all (on-chain) scaling solutions is horizontal scaling. This particular characteristic of sharding may make it the ideal technique for scaling up digital forensic for autonomous vehicles, due to their inherent physical nature. We propose horizontal scaling based on geographical boundaries of states within nations, because it is unlikely for a car crash in one state to be concerned about a car crash in another. Hence, this application scenario is the perfect fit for scaling by sharding due to limited cross-shard communication.

Ensuring correctness of data source. Blockchain as a kind of time stamping service cannot itself ensure the trustworthiness of facts, which originate "off-chain". The mechanism we discussed in this work ensures irreversibility of submitted data while it trusts the data sources. However, a method to check the correctness of the data would strengthen this mechanism. For instance, Physical Unclonable Functions [12] can be utilized to ensure the genuineness of controllers such as the Event Data Transmitters.

Clock Synchronization. Clock synchronization aims to coordinate otherwise independent clocks. Even when initially set accurately, real clocks will differ after some amount of time due to clock drift, caused by clocks counting time at slightly different rates. As a result two vehicles on the road may not share the same clock. There are several problems may arise in digital dorensics due to clock rate differences, e.g., cross-verification of data submitted on the blockchain.

Data Storage While the proposed "off-chain" data storage scheme resolves data storage limitations of the blockchain system itself, it however weakens data availability requirements, since the availability of this data now depends on the individual storage and shared counterparts. There is currently no mechanism for ensuring availability of critical forensic data on blockchain with the data stored in distributed file systems.

5. Conclusions.

In this paper, we systematically investigated trustable dispute and liability resolution on the blockchain in the event of traffic collisions or environmental accidents. We listed the involved organizations and the key requirements that must be met for trustable liability resolution and successful adoption of the system in practice. We considered permission-based blockchains to achieve relatively higher throughput. However, plagued by scaling issues in the number of nodes participating in the consensus, we proposed a novel consensus protocol that emulates influence spheres in the real-world. It conducts transaction validation in carefully constructed quorums, and ensures transaction immutability through submitting those transactions on a global blockchain. Then, the participants together ensure integrity by agreeing on the validity of one anothers transactions. We also discussed ways to enable privacy-aware autonomous driving on the blockchain and techniques to reduce data storage cost on the blockchain by utilizing off-chain peer-to-peer distributed file systems. Finally, we discussed various limitations of the proposed protocol and suggested some ways to remedy those concerns.

While the proposed blockchain technologies are relatively new and untested, we hope that our mapping of outstanding scalability, security and privacy limitations provides a useful starting point for a thorough simulation in this application domain.

REFERENCES

[1] *IEEE Standard for Wireless Access in Vehicular Environments 1609.12-2016*. Accessed: 03-22-2018.

[2] *Interplanetary file system (ipfs)*. https://www.theverge.com/transportation/2018/4/19/17204044/tesla-waymo-self-driving-car-data-simulation. 2018 (accessed April 27, 2018).

[3] *Neo consensus protocol*. docs.neo.org/en-us/node/consensus.html. 2018 (accessed May 4, 2018).

[4] N. H. Ab Rahman, W. B. Glisson, Y. Yang, and K.-K. R. Choo, *Forensic-by-design framework for cyber-physical cloud systems*, IEEE Cloud Computing, 3 (2016), pp. 50–59.

[5] F. Armknecht, G. O. Karame, A. Mandal, F. Youssef, and E. Zenner, *Ripple: Overview and outlook*, in International Conference on Trust and Trustworthy Computing, Springer, 2015, pp. 163–180.

[6] C. Cachin, *Architecture of the hyperledger blockchain fabric*, in Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016.

[7] M. Castro, B. Liskov, et al., *Practical byzantine fault tolerance*, in OSDI, vol. 99, 1999, pp. 173–186.

[8] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, *Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles*, arXiv preprint arXiv:1802.00561, (2018).

[9] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, *Blockbench: A framework for analyzing private blockchains*, in Proceedings of the 2017 ACM International Conference on Management of Data, ACM, 2017, pp. 1085–1100.

[10] L. Floridi, *Faultless responsibility: on the nature and allocation of moral responsibility for distributed moral actions*, Phil. Trans. R. Soc. A, 374 (2016), p. 20160112.

[11] J. H. Hartman, I. Murdock, and T. Spalink, *The swarm scalable storage system*, in Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on, IEEE, 1999, pp. 74–81.

[12] C. Herder, L. Ren, M. van Dijk, M.-D. Yu, and S. Devadas, *Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions*, IEEE Transactions on Dependable and Secure Computing, 14 (2017), pp. 65–82.

[13] Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu, *Senate: A permissionless byzantine consensus protocol in wireless networks*, arXiv preprint arXiv:1803.08694, (2018).

[14] S. Karnouskos and F. Kerschbaum, *Privacy and integrity considerations in hyperconnected autonomous vehicles*, Proceedings of the IEEE, 106 (2018), pp. 160–170.

[15] S. King and S. Nadal, *Ppcoin: Peer-to-peer crypto-currency with proof-of-stake*, self-published paper, August, 19 (2012).

[16] L. Lamport et al., *Paxos made simple*, ACM Sigact News, 32 (2001), pp. 18–25.

[17] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian, *Log your car: The non-invasive vehicle forensics*, in Trustcom/BigDataSE/I SPA, 2016 IEEE, IEEE, 2016, pp. 974–982.

[18] J. McPherson, *How Uber's Self-Driving Technology Could Have Failed In The Fatal Tempe Crash*, 2018 (accessed April 27, 2018), https://www.forbes.com/sites/jimmcpherson/2018/03/20/uber-autonomous-crash-death/#73e0a0c17fbe.

[19] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, (2008).

[20] S. O'Kane, *How tesla and waymo are tackling a major problem for self-driving cards: Data*. https://www.theverge.com/transportation/2018/4/19/17204044/tesla-waymo-self-driving-car-data-simulation. Accessed: 2018-04-29.

[21] D. Ongaro and J. K. Ousterhout, *In search of an understandable consensus algorithm.*, in USENIX Annual Technical Conference, 2014, pp. 305–319.

[22] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn, *A security credential management system for v2v communications*, in Vehicular Networking Conference (VNC), 2013 IEEE, IEEE, 2013, pp. 1–8.