IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Proof-of-Event Recording System for Autonomous Vehicles: A Blockchain-based Solution

HAO GUO[1,2], WANXIN LI[3], MARK NEJAD[3], and CHIEN-CHUNG SHEN[2]

[1]School of Software, Northwestern Polytechnical University, Taicang Campus 215400, China
[2]Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA
[3]Department of Civil and Environmental Engineering, University of Delaware, Newark, DE 19716, USA

Corresponding authors: Mark Nejad (e-mail: nejad@udel.edu).

This paper is a revised and extended version of [1] presented at the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN) in Shenzhen, China.

ABSTRACT Autonomous vehicles are capable of sensing their environment and navigating without any human inputs. However, when autonomous vehicles are involved in accidents between themselves or with human subjects, liability must be indubitably decided based on accident forensics. This paper proposes a blockchain-inspired event recording system for autonomous vehicles. Specifically, we design the mechanism of "Proof-of-Event" with a dynamic federation consensus to achieve indisputable accident forensics by providing trustable and verifiable event information. We propose a dynamic federation consensus scheme to verify and confirm the new block of event data in an efficient way without any central authority. We conduct numerical analyses and prototyped experiments based on the proposed fast leader election algorithm and the Hyperledger Fabric blockchain network. The results show that our system is effective and feasible in generating and storing accident records in blockchain-based vehicular networks. The security capability of the proposed scheme is also discussed against multiple threats and attack scenarios.

INDEX TERMS Autonomous vehicle, blockchain, connected vehicular network, data integrity and privacy, event recording, Hyperledger Fabric, leader election.

I. INTRODUCTION

Autonomous vehicles (also known as self-driving vehicles) are capable of navigating without any human input [2]. To facilitate self-driving, autonomous vehicles sense their surroundings via a variety of sensory technologies (such as lidar, camera, and ultrasound) and use a control system to interpret such sensory information to compute navigation paths, avoid obstacles, and follow traffic signs [1].

However, with autonomy, comes accountability. When autonomous vehicles are involved in accidents (collisions between themselves, or collisions with conventional vehicles, pedestrians or other objects), how could such events be automatically and reliably recorded for forensic purposes to determine liability? In addition, how could such recorded events be trusted, verified, and not tampered? Such issues become critical when there exist incentives for the different parties involved to tamper with the recorded events to avoid punitive penalties. This paper describes a blockchain-inspired event recording scheme which uses the proposed Proof-of-Event with Dynamic Federation Consensus to incorporate autonomous vehicles into a tamper-proof and verifiable event recording and forensics system.

A blockchain consists of a series of blocks, each of which is composed of sets of timestamped transactions and a hash of its previous block [3]. The idea of blockchain was first proposed in Bitcoin which solves the double-spending problem by using *Proof-of-Work* (PoW) mechanism [4]. In PoW, miners compete to become the winner of solving a hash puzzle so as to obtain the right for generating the next block and receiving incentives. However, PoW usually takes almost 10 minutes to solve such a puzzle and generate a new block. Due to the computational difficulty of PoW, miners tend to form bigger mining pools to conduct PoW [5], which diminishes one of the original Bitcoin features of being decentralized.

In an event recording system, accidents are recorded as timestamped transactions to be saved into a new block in real-time. Although autonomous vehicles may be equipped with reasonable computing capacity, conducting PoW to record the event of an accident in real-time will not be feasible due to the complexity and the time taken for solving a hash puzzle.

To address this critical issue, we propose the mechanism of Proof-of-Event with Dynamic Federation Consensus to record accident events in a new block. There are three types of participants in our blockchain network (Accident, witness, and verifier vehicles). Infrastructure participants (e.g., road-side units) serve as the verifier role, whereas autonomous vehicles can serve as either the accident witness or verifier roles. When an accident occurs, vehicles directly involved in the accident broadcast 'event generation' requests (via IEEE 802.11p [DSRC], for instance), which only those vehicles within the (DSRC) communication range will receive and respond. Then, the vehicles directly involved in the accident and those vehicles receiving the 'event generation' requests will generate and broadcast the event into a 'vehicular network' which is implemented based on the existing cellular network infrastructure. Within the vehicular network, a federated group of vehicles is dynamically formed and the lead verifier vehicle selected by a fast leader election algorithm will record the event data into a new block by using a multi-signature scheme [6], [7]. The generated new block may be made available to the agencies such as the Department of Motor Vehicles (DMV), for the permanent records.

The mechanism of Proof-of-Event with Dynamic Federation Consensus records events for indisputable accident forensics. It provides data integrity and trustworthiness by relying on the generated hash digest and utilizing event data from multiple sources. The recorded events also provide traceable evidence. Specifically, the proposed Dynamic Federation Consensus scheme replaces the role of PoW in the Bitcoin blockchain to confirm and record a new block in a fast and effective manner without incurring extensive computation. Since a federation is dynamically formed around each accident over a vehicular network, the consensus on the authenticity of the generated events can be recorded in a flexible and robust manner [1].

This paper makes the following contributions:

- We proposed an event recording system for vehicular networks. The system provides autonomous vehicles in a novel design solution inspired by the blockchain technology that maintains vehicular accident events as permanent digital records.
- Specifically, we proposed the mechanism of Proof-of-Event with Dynamic Federation Consensus based on an n-of-m voting scheme to record accident events into the newly generated transaction and saved in the blockchain. Also, a federated group is dynamically formed and the lead verifier can be selected based on fast lead verifier election algorithm. Our proposed scheme provides data integrity and trustworthiness by utilizing recorded accident event data from multiple sources.
- As a proof of concept, we developed a prototype of proposed architecture on local deployed Hyperledger Fabric [8] and Hyperledger Composer [9]. We also utilized the benchmark tool Hyperledger Caliper [10] to measure the performance of blockchain network. In addition, we conducted numerical analyses of the proposed fast leader election algorithm. The result showed that the mechanism is feasible and can be applied in real-world applications.

The remainder of the paper is organized as follows. Related work is described in Section II. In Section III, we present the cellular network-based vehicular network and describe the mechanism of Proof-of-Event with Dynamic Federation Consensus. In addition to normal cases, 'extreme' accident scenarios are discussed in Section IV. In Section V, we conduct extensive experiments for the proposed model and blockchain-based system. In Section VI, we analyze and discuss the proposed system against potential attacks. Section VII concludes the paper.

II. RELATED WORK

A. EVENT DATA RECORDERS

An event data recorder (EDR), a vehicle-equivalent of a plane's flight recorder or "black box," is installed in vehicles to record information related to crashes or accidents [11]. Some EDRs continuously record data until a crash or accident stops them, and others are activated by crash-like events (such as a sudden decrease in velocity) and may continue to record until the accident is over, or until the recording time expires [11]. Due to its individual and independent installation, once an EDR is damaged or malfunctions, there is no chance to restore or verify the information stored.

Heijden et al. [12] proposed a distributed ledger that provides accountability for both misbehaving authorities and vehicles. The goal is to reduce the requirements of trust in users of vehicular communication systems and to create accountability for misbehavior authorities via hierarchical consensus and global revocation. Cebe et al. [13] proposed a permissioned blockchain framework to manage the collected vehicle-related data. Specifically, they integrated vehicular public key infrastructure (VPKI) to the proposed blockchain system to provide membership establishment and privacy.

In contrast, our work focuses on accident forensics for autonomous vehicles. By employing the mechanism of Proof-of-Event with Dynamic Federation Consensus, accident events are stored in a trustable, verifiable, and tamper-proof manner [1].

B. BLOCKCHAIN-BASED VEHICULAR SYSTEMS

Yuan and Wang [14] proposed a decentralized blockchain-based intelligent transportation systems with usage of infrastructures and resources. As a case study, the authors describes a blockchain-based real-time ride-sharing system. By using Ethereum's smart contracts, Leiding et al. [15] proposed a self-managed and decentralized system to deploy and run different applications on vehicular ad-hoc networks without a central managing authority. Rowan et al. [16] proposed an inter-vehicle session key establishment protocol to secure vehicle-to-vehicle communications through visible light and acoustic side-channels. Luo et al. [17] proposed blockchain enabled trust-based location privacy protection scheme in VANET, and devised the trust management method based

on the Dirichlet distribution, meaning that both the requester and the cooperator will only cooperate with trusted vehicles. Baza et al. [18] proposed a privacy-preserving and trustless ridesharing application with proof of concept implemented atop the Ethereum network. Li et al. [19] proposed a blockchain and zero-knowledge proof inspired approach to address the data integrity and privacy issue in traffic management. Sharma et al. [20] proposed a blockchain-based vehicle network architecture in smart cities to facilitate the construction of transport management systems and other transportation scenarios. However, none of the work took inspiration from the blockchain to address the activity of accident forensics for autonomous vehicles.

C. CONSENSUS MECHANISMS

As consensus is critical to the decentralized nature of blockchain, we review existing consensus schemes to highlight our unique contribution. In the current state-of-the-art, PoW [21], Proof of Stake (PoS) [22], Proof of Authority (PoA) [23], and several other Proof of 'X' consensus models all rely on selecting one single peer to produce the new block. For instance, PoW selects one single peer by "nonce lottery" via mining, PoS randomly selects a peer among the largest stakeholders [24], and PoA is a modified form of PoS where a validator's identity serves as the role of stake. However, these consensus models gradually deviate from the original goals of decentralization and democratization. For instance, large mining pools coordinate authorities of Bitcoin, PoS concentrates power in the hands of few peers based on their balances, and PoA leaves the decision of which entities can generate new blocks to one central authority [24]. In contrast, the mechanism of Proof-of-Event with Dynamic Federation Consensus addresses the dynamic and autonomous nature of self-driving vehicles so that the accident forensic information could be validated by a dynamically formed federation of the vehicles [1].

D. DISTRIBUTED LEADER ELECTION ALGORITHMS

In distributed system, leader election is the process of designating one single node as the coordinator of some tasks distributed among multiple nodes, where nodes communicate among one another to decide which of them will become the leader [25]. Typically, leader election algorithms assume that every node in the system has a unique priority number (for instance, an ID), and the node with the highest priority will be elected as the leader. When the existing leader fails, a leader election algorithm reelects the node which now has the highest priority number. We review two leader election algorithms, the Bully algorithm [26] and the Ring algorithm [27].

In the Bully algorithm, there are three types of messages: (1) *Election*, which starts the election; (2) *Answer*, which acknowledges a message; (3) *Leader*, which declares a leader. Assuming that each node knows the IDs of every other nodes when the system is initialized, the node with the "highest" ID becomes the current leader by default. If any node detects the failure of the current leader, it will wait for a timeout period and then restart the election process as follows. The node sends the *Election* message only to the nodes with higher IDs than itself. If no one replies after a timeout period, this node declares itself as the winner and starts acting as the leader. Otherwise, if other nodes reply with *Answer* messages, this node will wait for the *Leader* message. In the worst-case scenario when the node with the lowest ID detects the failure of the leader, the message complexity of the Bully algorithm is $O(N^2)$.

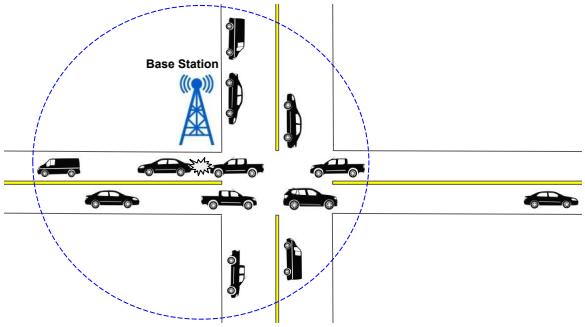


FIGURE 1. Cellular network-based 'vehicular' network of the accident.

In the Ring algorithm, nodes are organized in a logical ring and all the messages are sent around the ring. There are two types of messages: (1) *Election*, which starts the election; (2) *Leader*, which declares a winner. A node initiates an election upon detecting that the current leader has failed, by sending out an *Election* message with its own ID. An *election* message is forwarded around the ring. When a node receives the *Election* message [election, this-ID]:

- If this-ID > receiver's ID, it forwards the *Election* message and set its state to active participating.
- If this-ID < receiver's ID, it sends the updated *Election* message [election, receiver's ID] and set its state to active participating.
- If this-ID == receiver's ID, it broadcasts the *Leader* message.

In the worst case where the Ring algorithm operates in a system of N nodes, sends $N-1$ *Election* messages to first reach the new leader, another N *Election* messages to confirm that itself has been elected as the leader, and yet another N *Leader* messages to announce itself to be the leader. In total, the message complexity is $O(3N-1)$.

III. ARCHITECTURE OF EVENT RECORDING SYSTEM

A. CELLULAR NETWORK-BASED VEHICULAR NETWORK

For each accident, we use a cellular network-based infrastructure to define a 'vehicular network' where all the vehicles that are served by the same base station (i.e., within the same cell) of the vehicle(s) directly involved in the accident belong to the vehicular network as depicted in Fig. 1. We also assume that all the autonomous vehicles registered their license plate and VIN number with an authority, such as the Department of Motor Vehicles (DMV).

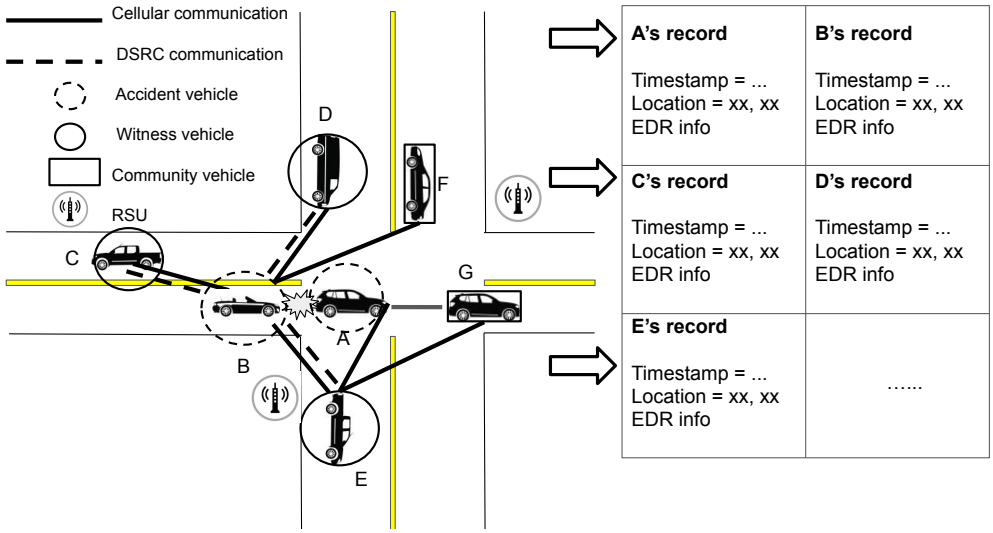


FIGURE 2. After an accident, both accident and witness vehicles generate and broadcast event data.

Vehicles use the IEEE 802.11p standard of Dedicated Short-Range Communications (DSRC) [28] to send and receive 'event generation' requests. Meanwhile, vehicles are connected to the cellular network to broadcast and confirm event data within the corresponding vehicular network.

B. PROOF-OF-EVENT WITH DYNAMIC FEDERATION CONSENSUS

To facilitate forensic investigation after an accident, one critical issue is the correctness and trustworthiness of the recorded event data, as vehicles involved, both directly and as bystanders, might be incentivized to alter accident-related information to avoid punitive penalties. Therefore, it is crucial to record authenticated event data at the specific time and location of the accident so that the recorded accident information could later be retrieved and cross-examined to determine liability. We propose the following two steps to accomplish the goal: first, to gather trustable event data from both vehicles directly involved in the accident and neighboring vehicles, then to verify and save the event data with the help of a dynamically formed federation of vehicles within the same vehicular network.

1) Gathering event data

Vehicles directly involved in an accident are termed "accident" vehicles, vehicles within the DSRC transmission range from the accident scene are termed "witness" vehicles, and vehicles within the vehicular network but outside the DSRC transmission range from the accident scene are termed "community" vehicles. To record the event of an accident, upon the occurrence of an accident, "accident" vehicles broadcast 'event generation' requests to "witness" vehicles. Fig. 2 depicts a scenario, in which "accident" vehicles A and B collided in an accident. "Witness" vehicles C, D, and E within the DSRC transmission range from the accident scene receive the event generation requests and confirm with the

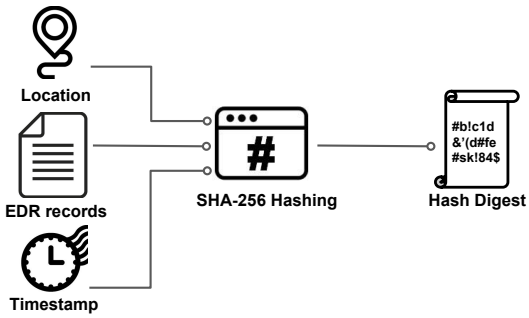


FIGURE 3. Hash digest of accident event data.

"accident" vehicles via DSRC [1]. Nearby Roadside Units (RSU) receiving such requests may also participate.

Then, both "accident" and "witness" vehicles generate their respective event data of timestamps, location, and EDR records (which contains histories of sensor readings, such as speed and steering angle, up to the moment of accident and around the accident scene), together with the corresponding hash digest as illustrated in Fig. 3, and broadcast their event data via cellular communications within the vehicular network. EDR serves as the origin evidence of the accident [1]. All the broadcast event data from both the "accident" and "witness" vehicles will be verified and saved in a new block by the lead verifier of a dynamically formed federation of vehicles and RSUs to be described in the next subsection.

2) Verifying and creating new block of accident event

Upon the occurrence of an accident, "accident" vehicles also broadcast, via cellular communications, 'federation formation' requests to the "community" vehicles in the vehicular network to start the selection of a subset of "community" vehicles as "verifier" vehicles to form a federation. To reduce communication overhead, we adopt a self-selection process where each vehicle has a reputation score, which is deter-

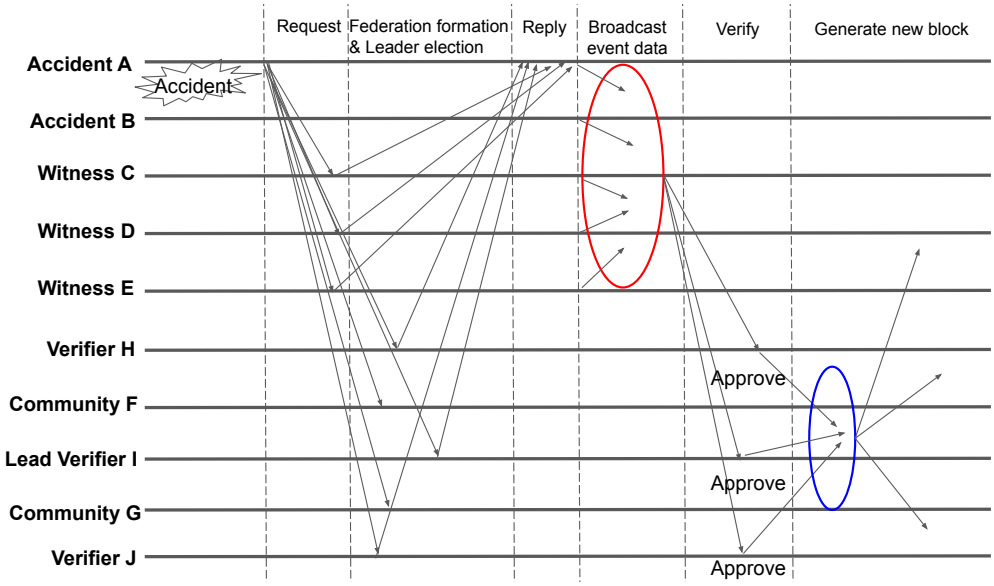


FIGURE 4. Sequence diagram for accident, witness, community, and verifier vehicles.

mined based on a vehicle's driving and reporting records, so that a vehicle having a reputation score higher than a pre-defined threshold becomes a "verifier" and responds to the "accident" vehicle with its reputation score. Then, the "verifier" vehicle with the smallest delay time will be designated as the lead verifier via a distributed lead verifier election algorithm to be described next, who is responsible for generating a new block for the accident and sending the block to the DMV to be inserted into the blockchain.

We propose the Fast Lead Verifier Election algorithm inspired by the Bully algorithm while reducing the overheads of communication and confirming the leader in a more efficient way. The election starts by the "accident" vehicle broadcasting an *Election* message, containing its GPS coordinates and all the IDs of the "verifier" vehicles, to all the "verifier" vehicles so that each "verifier" vehicle knows about other "verifier" vehicles. Inspired by the Timer-Based-Best-Select (TBBS) scheme [29], [30], we define the delay response time $delay_i$ for verifier vehicle i as:

$$delay_i = \frac{1}{\eta} \times \frac{\lambda(\omega_1 d_i)}{\omega_2 r_i}. \tag{1}$$

In Eq. 1, η is the type parameter that indicates different weights of entities such as autonomous vehicles and RSUs. For instance, RSUs have a higher weight η compared with the autonomous vehicles. λ is the system parameter, d_i is the distance from the "accident" vehicle to "verifier" vehicle i with the weight ω_1, and r_i is the reputation score for "verifier" vehicle i with weight ω_2. However, Eq. 1 has one potential drawback. If there are multiple vehicles having the same reputation score r_i and the same distance d_i, all these vehicles will have the same $delay_i$ by applying Eq. 1.

Due to the possibility of generating the same time delay, we enhance the Eq. 1 as:

$$delay_i = \frac{1}{\eta} \times \frac{\lambda(\omega_1 d_i)}{\omega_2 r_i} + \tau, \tag{2}$$

where τ is the random generated number which follows the uniform distribution $\tau \sim U(0, 10ms)$. After adding the τ parameter, $delay_i$ will yield a different result even with the same reputation score r_i and the same distance d_i for each "verifier" vehicle i. In the extreme scenarios when the delay response time result is still the same, the leader will be randomly selected among the candidates. In Algorithm 1, the vehicle who received the *Election* message will reply a unique delay response time $delay_i$. After receiving the minimum $delay_i$, the "accident" vehicle who starts the election will confirm the "verifier" vehicle i as the new leader. Finally, the elected vehicle i can broadcast the leader (victory) status.

As depicted in Fig. 4, after the "accident" and "witness" vehicles generate and broadcast event data into the vehicular network, "verifier" vehicles take the responsibility of validating the received event data against the received hash digests, and confirm with the lead "verifier" vehicle (I in Fig. 4). The lead "verifier" vehicle executes the n-of-m voting scheme to achieve federation consensus when n out of m "verifier" vehicles confirm, and generates a new block of accident event. The lead "verifier" vehicle may then broadcast the new block to all the "community" vehicles. In our proposed scheme, we argue that RSU is usually trustable and honest participant node. If the accident location has the nearby RSU, based on the close geographical location d_i and high weight parameter value η, the RSU could serve as the lead verifier.

Compared to PoW in Bitcoin network, our solution does not incur any expensive computation associated with mining. Unlike PoA, which relies on the decision of one single authority, our solution demands confirmations from multiple authorities, if the n-of-m vote threshold is satisfied, verifier

Algorithm 1 Fast Lead Verifier Election Algorithm

OUTPUT: The leader of verifier federation
START UP *Accident vehicle A starts the Election.*
A sends $Election$ message to all verifier vehicles P_i;
A waits for messages in a time period T;
A gets $answer$ message from all P_i;
▷ Answer has delay response time $delay_i$
if no $answer$ within time T **then**
    A becomes the Leader; ▷ A is the new leader.
    A sends a $Leader$ message to all verifier vehicles;
    A stops $Election$ process;
end if
END START UP

UPON EVENT *Accident vehicle A receives the Answer message from a verifier vehicle:*
A computes the running minimum delay response time $mindelay$ and broadcasts to all verifier vehicles after receiving $Answer$ messages from all verifier vehicles;
END UPON EVENT

UPON EVENT *Accident vehicle A receives the Leader message:*
A accepts the sender with minimum delay response time as the $Leader$;
A stops the $Election$;
END UPON EVENT

UPON EVENT *Verifier vehicle P_i receives the Election message:*
P_i calculates $delay_i = \frac{1}{\eta} \times \frac{\lambda(\omega_1 d_i)}{\omega_2 r_i} + \tau$;
P_i sends $answer$ message containing $delay_i$ to A;
END UPON EVENT

UPON EVENT *Verifier vehicle P_i receives the Answer message containing $mindelay$ from A:*
if P_i has $mindelay$ **then**
    P_i becomes the $Leader$ and broadcasts the $Leader$ message to verifier vehicles;
else
    P_i waits $Leader$ message from other verifier vehicle;
end if
END UPON EVENT

UPON EVENT *Verifier vehicle P_i receives the Leader message from P_j:*
P_i accepts other verifier vehicle P_j as the new $Leader$;
END UPON EVENT

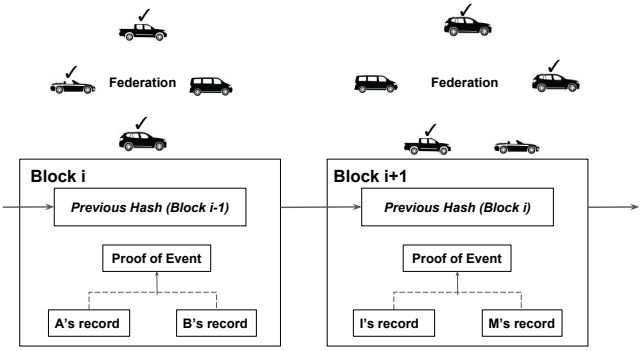


FIGURE 5. Each block contains event data and hash value of the previous block and is confirmed by the verifier vehicles from the federation. The blockchain is maintained by the DMV.

vehicles approve the event records and generate a new block. Every new block is linked with the previous block by the hash header. For instance, as depicted in Fig. 5, block i is verified by 3 out of 4 verifier vehicles from the federation, while the block i+1 is verified by the 3 out of 5 cases. Finally, all the newly generated blocks of accident event will be saved permanently in DMV to keep the record.

Note that "witness" vehicles (C, D, and E) function differently from "verifier" vehicles (H, I, and J). The job of the former is to generate event data, while that of the latter is to verify event data and generate a new block of the accident event. "Witness" vehicles are close to the accident scene, whose EDR records may contain sensory readings related to the "accident" vehicles. In contrast, "verifier" vehicles are dynamically chosen, which are located at random geographical locations within the same cell, even away from the accident scene, which makes them more neutral and independent. The decoupling of event data generations from their verification process mitigates the possibility of any malicious activities, such as tampering of event data and collusion among vehicles [1].

As we know from the leader election process of Bully algorithm, the time complexity in worst case is $O(N^2)$. In contrast, our proposed fast leader election algorithm will include the minimum response delay time in the $answer$ message, and only the autonomous vehicle who has the minimum delay time $delay_i$ will be elected as the $Leader$. In contrast, the time complexity is $O(N)$ in our algorithm. As shown in Table 1, the total number of messages in our proposed Algorithm 1 is significantly less than the Bully algorithm.

TABLE 1. Number of Messages Passing

# of Nodes	Bully Algorithm	Our Algorithm
10	100 messages	20 messages
50	2500 messages	100 messages
100	10000 messages	200 messages

C. INCENTIVES FOR PARTICIPATION AND HONESTY

Bitcoin supplies new bitcoins to miners as incentives for their efforts of PoW [4]. However, there is no tangible incentives in the proposed Proof-of-Event. To motivate autonomous vehicles to participate as either "witness" or "verifier", different incentives (or rewards) must be defined. For instance, being a "witness" or "verifier" could raise a vehicle's credit score and lower its insurance premium. Also, "accident" vehicles that engage reliably in Proof-of-Event and cooperate fully in accident forensics may receive a reduced liability.

1) Design of Incentive Mechanism

In our proposed system, an honest lead verifier should be rewarded for conducting positive behavior. To increase the probability of an honest verifier being selected as the leader, we designed the following incentive mechanism:

$$C_i^R = C_i + R_i, \tag{3}$$

$$R_i = \sum_{j=1}^{k-1} (\psi_j \times \zeta_j) + (\psi_k \times r_c), \tag{4}$$

$$\zeta_j = \sum_{c=1}^{j-1} (r_c). \tag{5}$$

In Eq. 3, C_i^R is the most updated reputation score of the verifier i. C_i is the initial reputation score of verifier i. R_i is the reward function which is defined in Eq. 4. We assign different reward weights according to the time of the reported accident instances. The closer to the current time period, the larger the weight ψ_j. As a result, ψ_k has the highest weight in our proposed scheme. This can guarantee that reward is mainly determined by the cumulative positive behavior in the most recent time. Assume that verifier i had participated in $k-1$ instances of accident in the past. The first term of Eq. 4 represents all previous accidents in the history, and k denotes the current instance of accident. ζ_j defined in Eq. 5 is the cumulative score of behavior in previous history. $r_c = 1$ if there is positive behavior in a given instance of an accident, otherwise, $r_c = -1$.

D. REVIEWING BLOCKS FOR ACCIDENT FORENSICS

Later, people (police or judge) can review the accident event data stored in the blockchain from the DMV's record. If there is no discrepancy between event data generated by "accident" and "witness" vehicles, liability can be clearly determined. Otherwise, further investigation becomes necessary. For instance, in Fig. 2, if "accident" vehicle B reported its speed as 20 mph, while other "witness" vehicles (C, D, and E) reported higher speeds for B, it is highly likely that "accident" vehicle B had a faulty speed sensor which caused it to speed and collided with vehicle A.

IV. EXTREME SCENARIOS

Our proposed mechanism works the best in accident scenarios where the density of the cellular-based vehicular network covering the accident scene is above a certain threshold. In such cases, there are enough "witness" to generate event data vehicles and enough "verifier" vehicles to form a federation, reach a consensus, and create a new block. However, there exist the following three 'extreme' scenarios when such vehicular network is very sparse or no vehicle around the accident scene [1].

(1) Neither "witness" nor "verifier" vehicle exists for an accident scene. Since there is no "witness" or "verifier" vehicle, no new block could be generated. The EDRs of the "accident" vehicles will be the only evidence for future forensic investigation [1].

(2) No "witness" vehicle exists for an accident scene. A new block will be created based on the event data generated only by the "accident" vehicles [1].

(3) No "verifier" vehicle exists for an accident. In this case, there exists (few) "witness" vehicles around the accident scene, but no "verifier" vehicle within the vehicular network. We argue that such scenarios will be rare in practice [1].

V. EXPERIMENTS AND EVALUATION

A. NUMERICAL ANALYSIS

In Section III, we proposed the fast leader election algorithm. The parameters used in the simulation are shown in Table 2.

TABLE 2. Parameter Setting

Parameter Name	Value (Range)
Number of autonomous vehicles N	1-100
System Parameter λ	1-3
Type Parameter η	1-3
Distance Value d_i	1-100m
Weight Parameter ω_1	0.5
Weight Parameter ω_2	1
Reputation Score r_i	1-100
Initial RSU Score r_{rsu}	20
Initial AV Score r_{av}	10
RSU Weight Parameter ψ_{rsu}	10
AV Weight Parameter ψ_{av}	5
Random Generated Number τ	0-10ms

The minimum delay (seconds) $delay_i$ results are shown in Fig. 6. As we can observe from the Eq. 1 and Eq. 2 proposed in Section III, the minimum delay time increases when the system parameter λ increases. Also, with the increased range of distance value in the system, the minimum delay of $delay_i$ also increases significantly.

The probability of the same delay time P_{dt} results are shown in Fig. 7. The ratio of the probability of the same delay time increases dramatically when the reputation scores r_i has a relatively small range (R = 5 and R = 10). This is because within the small range of reputation score r_i, the result of the same delay time will have a higher chance to overlap with each other. While within the large range of reputation score r_i (R= 50 and R = 100), the probability of same delay time P_{dt} increases slightly.

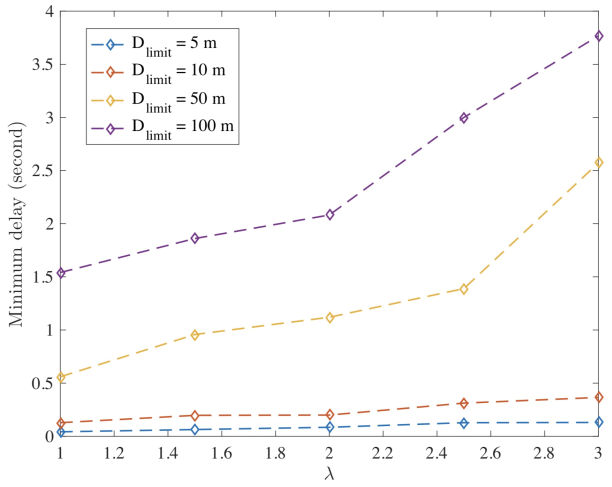


FIGURE 6. The minimum delay result.

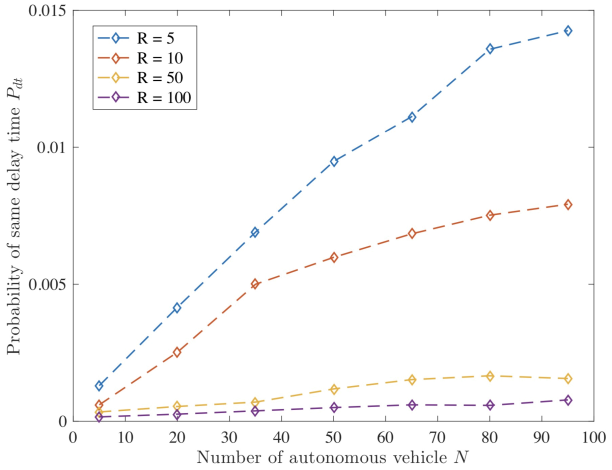


FIGURE 7. The probability of same delay time result.

The cumulative reputation score results for RSU and autonomous vehicle are shown in Fig. 8. In this experiment, we conduct 5 rounds of accident events with a setting of positive behaviors $r_c = 1$ for both RSU and autonomous vehicle. The initial reputation score is 20 for RSU, and 10 for autonomous vehicle. The cumulative reputation score for RSU increases faster comparing to autonomous vehicle because RSU's weight parameter ψ_{rsu} has a higher value than autonomous vehicle's weight parameter ψ_{av}. In addition, the final reputation score is mainly determined by the most recent reward R_i since our proposed incentive mechanism guarantees that the reputation score is mainly determined by most recent accident behavior.

B. EXPERIMENT SETUP

We prototyped the proposed Proof-of-Event recording system and conducted a series of experiments to evaluate the performance. The blockchain module was developed on the Hyperledger Fabric platform and evaluated by the Hyperledger Caliper benchmark tool. These experiments were conducted

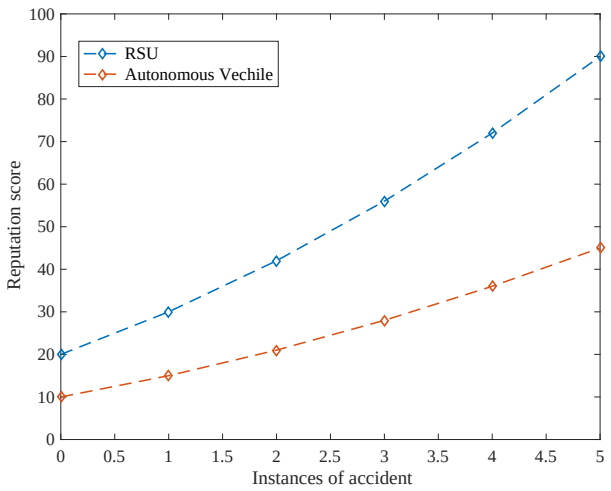


FIGURE 8. RSU's cumulative reputation scores vs. autonomous vehicle's cumulative reputation scores.

on a laptop running Ubuntu 18.04 operating system with 2.8 GHz Intel i5-8400 processor, 8GB of memory as the default settings.

C. BLOCKCHAIN NETWORK

As shown in Fig. 9, we develop the blockchain-inspired EDR network on Hyperledger Fabric platform, which maintains a distributed ledger for recording and sharing EDR data including ID number, owner, location, speed, steering angle and hash digest information. The data structures of participants are defined in the Model file (.cto), while the smart contract is written in the Script file (.js). We utilize the Hyperledger Composer to generate the unit file (.bna) and deploy it to the Fabric network. In our prototype system, it has 2 accident vehicles, 2 witness vehicles, 2 verifier vehicles, and 2 RSUs as initialized instances in the blockchain network.

Hyperledger Composer also provides a webpage interface for interacting with the blockchain network. Each participant

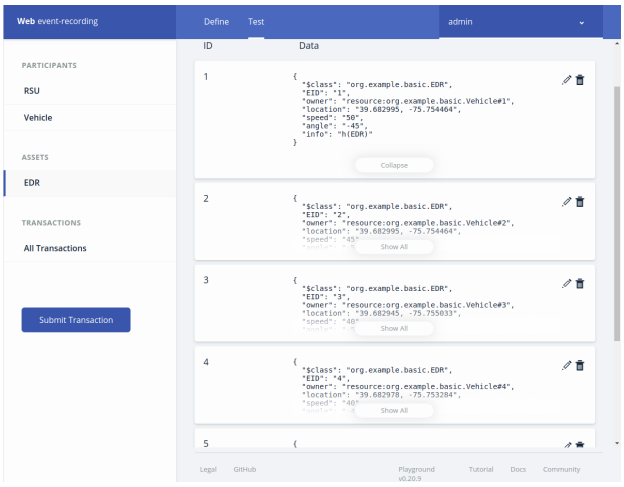


FIGURE 9. EDR data on blockchain network.

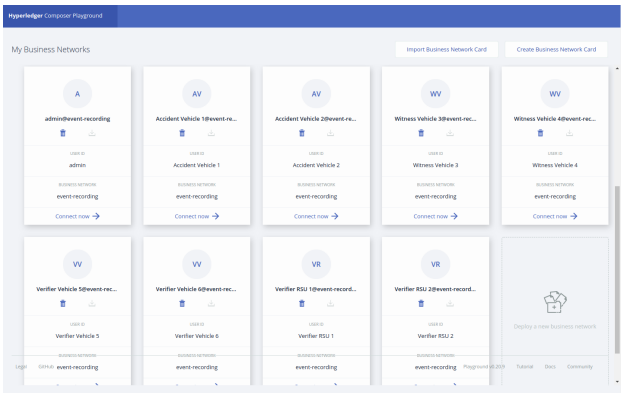


FIGURE 10. Blockchain-based EDR network login window.

has a unique ID mapping the identity in the blockchain network as shown in Fig. 10. Also, we utilize the Hyperledger Caliper, which is a blockchain benchmark tool, to measure the performance of a blockchain network implemented with a set of predefined use cases and parameters.

1) Success rate

The transaction success rate is related to endorsement policies. [1] For instance, the 2-of-6 policy is much resilient against malicious attackers when compare with the 1-of-6 policy. In the 1-of-6 policy, one compromised peer could influence the transaction processing, which may allow “fake transaction" into the ledger. All of the transaction testing cases have the 100% successful rate with different endorsement policies based on the results of our experiments.

2) Transaction throughput

We evaluate the transaction throughput results with different endorsement policies as shown in Fig. 11. Our blockchain network has 27.4 tps, 11.6 tps, and 8.6 tps under 1-of-6, 2-of-6, and 3-of-6 endorsement policy, respectively. As a result, when the system increases the number of peers participating in the endorsement process, the average transaction throughput will decrease significantly.

3) Transaction latency

Transaction latency measures the time for an issued transaction from being submitted to processed on the ledger. The experiment is configured based on different endorsement policies: 1-of-6, 2-of-6, and 3-of-6. As we can see from Fig.12, with the increasing number of endorsing peers, both the maximum, minimum, and average latency time also increase significantly.

4) Resource consumption

We measure the resource consumption with the leveldb provided by the Hyperledger Caliper for each validating peer under different endorsement policies. To be more specific,

[1]Hyperledger Fabric has the pre-configured network with 6 peers participating in the endorsement policy.

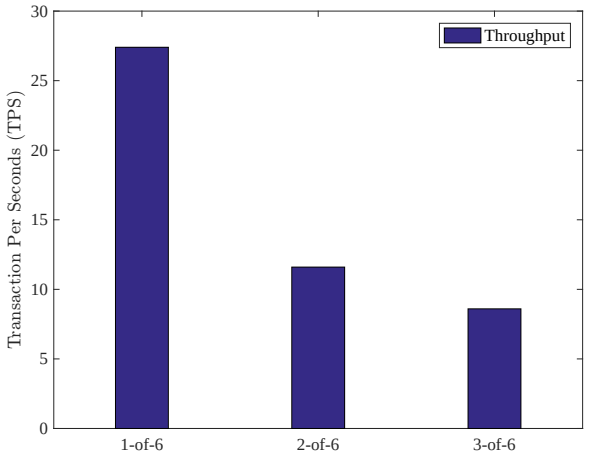


FIGURE 11. Transaction throughput with different endorsement policy.

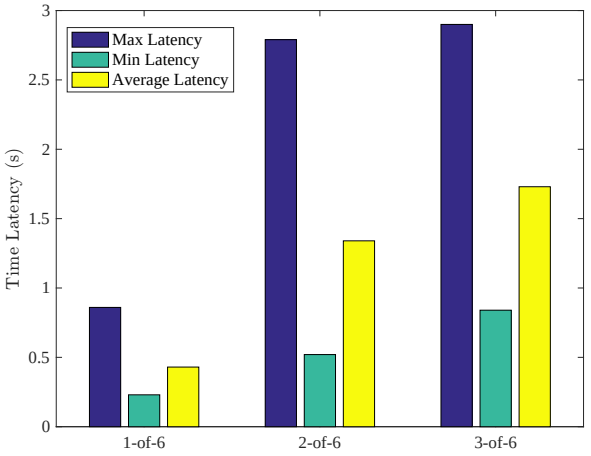


FIGURE 12. Transaction latency with varying endorsement policy.

the average cost of memory, CPU, traffic in, and traffic out data are recorded as the result. As shown in Table. 3, when the number of peers participating in endorsement policies increases, the total memory usage for each individual peer belonging to the endorsement policies also increases. For the CPU rate, we found the decreased usage rate for each individual peer when the system increases the peers participating in the endorsement policy. However, the traffic I/O speed decreases when we increase the number of validating peers in endorsement policies. Also, from the proposed fast leader election algorithm, we observed that the lead verifier vehicle will consume more memory usage and CPU rate.

TABLE 3. Resource Consumption

Type	Name	Memory	CPU	Traffic in	Traffic out
1-of-any	peer1	210MB	24.7%	2.1MB	1.4MB
2-of-any	peer2	171MB	15.6%	1.2MB	0.67MB
2-of-any	peer3	259MB	14.9%	1.3MB	0.72MB
3-of-any	peer4	200MB	9.1%	0.91MB	0.42MB
3-of-any	peer5	228MB	9.3%	0.95MB	0.46MB
3-of-any	peer6	259MB	9.4%	0.94MB	0.45MB

VI. DISCUSSION ON POTENTIAL ATTACKS

In this section, we discuss the robustness of the proposed event recording scheme with respect to potential attacks.

A. TAMPERING ACCIDENT EVENT DATA

Existing EDRs installed on individual vehicles may be hacked and tampered to avoid the possible liability. Our proposed Proof-of-Event with dynamic federation scheme includes both "accident" and "witness" vehicles generate the accident event data, which is to be validated by an independent group of "verifier" vehicles to avoid the possibility of collusion. The individually recorded event data in the block could be cross-examined later to determine cause and liability. Further, the use of DSRC communications range limits which vehicles could serve as a witness role, which prevents vehicles away from the accident scene to generate any 'fake' event data [1].

B. IMPERSONATION ATTACK

As mentioned in Section III-A, legitimate autonomous vehicles are required to register the identity with DMV. A malicious vehicle may impersonate a reputable vehicle so as to be selected as a "verifier" vehicle. Unless there is enough number of colluding vehicles selected within the same federation, the use of n-of-m voting scheme to approve a new block is to lower the possibility of invalidating consensus [1]. Also, based on the incentive mechanism proposed in III-C, if the autonomous vehicle conducts malicious behavior, the reputation score will reduce significantly which also decreases the possibility of the malicious vehicle being selected as the "verifier" vehicle.

C. FAKE WITNESS VEHICLE ATTACK

As mentioned before, 'event generation' requests are broadcast via DSRC so that only the nearby "witness" vehicles, which receive such requests, can generate the accident event data. However, a malicious "witness" vehicle might forward the 'event generation' request to other vehicles which are beyond the range of DSRC communication, and 'invite' them to respond. Such act may launch the fake "witness" vehicle attack, where fake "witness" vehicles generate the fake event data in favor of "accident" vehicles. One possible solution to prevent such attacks is to set a small time window and deadline for "witness" vehicles to reply and broadcast the newly generated accident event data [1].

VII. CONCLUSION

As autonomous systems are becoming essential parts of our life, proper systems must be put in place to "look after" them so as to determine liability from malfunctions, defects, or even malicious attacks. By drawing inspiration from blockchain, this paper presents a novel approach to providing a tamper-proof and verifiable event recording system for accident forensics of self-driving vehicles as they are the most influential autonomous systems in our society. We conduct numerical analyses based on the innovative "Proof-of-Event" mechanism with fast leader election algorithm. Also, we prototype a blockchain-inspired vehicular network on Hyperledger Fabric. The performance evaluated by the Hyperledger Caliper reveals that our proposed system is effective and feasible in generating and storing accident records in blockchain-inspired vehicular networks.

REFERENCES

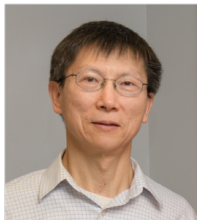
[1] H. Guo, E. Meamari, and C.-C. Shen, "Blockchain-inspired event recording system for autonomous vehicles," in 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN). IEEE, 2018, pp. 218–222, doi: 10.1109/HOTICN.2018.8606016.
[2] S. K. Gehrig and F. J. Stein, "Dead reckoning and cartography using stereo vision for an autonomous car," in Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on, vol. 3. IEEE, 1999, pp. 1507–1512.
[3] J. Dilley, A. Poelstra, J. Wilkins, M. Piekarska, B. Gorlick, and M. Friedenbach, "Strong federations: An interoperable blockchain solution to centralized third party risks," arXiv preprint arXiv:1612.05491, 2016.
[4] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2084–2123, 2016.
[5] I. Eyal, "The miner's dilemma," in Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 2015, pp. 89–103.
[6] https://en.bitcoin.it/wiki/Multisignature, 2019.
[7] H. Guo, W. Li, E. Meamari, C. C. Shen, and M. Nejad, "Attribute-based multi-signature and encryption for ehr management: A blockchain-based solution," in 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2020, pp. 1–5.
[8] "Hyperledger Fabric," 2020/02. [Online]. Available: https://www.hyperledger.org/projects/fabric
[9] "Hyperledger Composer," 2020/02. [Online]. Available: https://www.hyperledger.org/projects/composer
[10] "Hyperledger Caliper," 2020/04. [Online]. Available: https://www.hyperledger.org/projects/caliper
[11] H. C. Gabler, D. J. Gabauer, H. L. Newell, and M. E. O'Neill, "Use of event data recorder (edr) technology for highway crash data analysis," NCHRP Project, pp. 17–24, 2004.
[12] R. W. van der Heijden, F. Engelmann, D. Mödinger, F. Schönig, and F. Kargl, "Blackchain: Scalability for resource-constrained accountable vehicle-to-x communication," CoRR, vol. abs/1710.08891, 2017.
[13] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, "Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles," IEEE Communications Magazine, vol. 56, no. 10, pp. 50–57, 2018.
[14] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016, pp. 2663–2668.
[15] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular ad-hoc networks," in Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct. ACM, 2016, pp. 137–140.
[16] S. Rowan, M. Clear, M. Gerla, M. Huggard, and C. M. Goldrick, "Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels," arXiv preprint arXiv:1704.02553, 2017.
[17] B. Luo, X. Li, J. Weng, J. Guo, and J. Ma, "Blockchain enabled trust-based location privacy protection scheme in vanet," IEEE Transactions on Vehicular Technology, vol. 69, no. 2, pp. 2034–2048, 2019.
[18] M. Baza, N. Lasla, M. Mahmoud, G. Srivastava, and M. Abdallah, "B-ride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain," IEEE Transactions on Network Science and Engineering, 2019.
[19] W. Li, H. Guo, M. Nejad, and C. C. Shen, "Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach," IEEE Access, vol. 8, pp. 1–1, 2020.
[20] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-vn: A distributed blockchain based vehicular network architecture in smart city." Journal of Information Processing Systems, vol. 13, no. 1, 2017, doi: 10.3745/JIPS.03.0065.

[21] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in Proceedings of the IFIP TC6/TC11 Joint Working Conference on Secure Information Networks: Communications and Multimedia Security, ser. CMS '99. Deventer, The Netherlands, The Netherlands: Kluwer, B.V., 1999, pp. 258–272. [Online]. Available: http://dl.acm.org/citation.cfm?id=647800.757199
[22] P. Vasin, "Blackcoin's proof-of-stake protocol v2," URL: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper. pdf, 2014.
[23] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain," 2017.
[24] http://www.ubiquicoin.com/assets/proof.pdf, 2019.
[25] S. Vasudevan, J. Kurose, and D. Towsley, "Design and analysis of a leader election algorithm for mobile ad hoc networks," in Proceedings of the 12th IEEE International Conference on Network Protocols, 2004. ICNP 2004. IEEE, 2004, pp. 350–360.
[26] H. Garcia-Molina, "Elections in a distributed computing system," IEEE transactions on Computers, no. 1, pp. 48–59, 1982.
[27] G. N. Frederickson and N. A. Lynch, "Electing a leader in a synchronous ring," Journal of the ACM (JACM), vol. 34, no. 1, pp. 98–115, 1987.
[28] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 GHz DSRC-based vehicular safety communication," IEEE Wireless Communications, vol. 13, no. 5, 2006.
[29] V. Shah, N. B. Mehta, and R. Yim, "Optimal timer based selection schemes," IEEE Transactions on Communications, vol. 58, no. 6, pp. 1814–1823, 2010.
[30] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection," IEEE Journal on selected areas in communications, vol. 24, no. 3, pp. 659–672, 2006.

HAO GUO received the B.S. and M.S. degrees from the Northwest University, Xi'an, China in 2012, and the Illinois Institute of Technology, Chicago, United States in 2014, and his Ph.D. degree from the University of Delaware, Newark, United States in 2020, all in computer science. He is currently an Assistant Professor with the School of Software at the Northwestern Polytechnical University. His research interests include blockchain and distributed ledger technology, data privacy and security, cybersecurity, cryptography technology, and Internet of Things (IoT). He is a member of both ACM and IEEE.

WANXIN LI received the BSc and MSc degrees in computer science from the Chongqing University, Chongqing, China in 2015, and the University of Delaware, United States in 2017, respectively. He is currently working toward the Ph.D. degree at the University of Delaware. His research interests are in the area of blockchain, intelligent transportation systems (ITS), connected and autonomous vehicles, and Internet of Things (IoT). He is a member of IEEE.

MARK NEJAD is an Assistant Professor in the Department of Civil and Environmental Engineering at the University of Delaware. His research interests include connected and automated vehicles, network optimization, parallel and distributed computing, blockchain, and game theory. He has published more than thirty peer-reviewed papers in venues such as Transportation Science, IEEE Transactions on Parallel and Distributed Systems, and IEEE Transactions on Computers. He received several publication awards including the 2016 best doctoral dissertation award of the Institute of Industrial and Systems Engineers (IISE) and the 2019 CAVS best paper award from the IEEE VTS. He is a member of the IEEE and INFORMS.



CHIEN-CHUNG SHEN received his B.S. and M.S. degrees from National Chiao Tung University, Taiwan, and his Ph.D. degree from UCLA, all in computer science. He was a research scientist at Bellcore Applied Research working on control and management of broadband networks. He is now a Professor in the Department of Computer and Information Sciences of the University of Delaware. His research interests include blockchain, Wi-Fi, SDN and NFV, ad hoc and sensor networks, dynamic spectrum management, cybersecurity, distributed computing, and simulation. He is a recipient of NSF CAREER Award and a member of both ACM and IEEE.