Design of a Two Layered Blockchain-Based Reputation System in Vehicular Networks

Soojin Lee, *Student Member, IEEE*, and Seung-Hyun Seo, *Member, IEEE*

***Abstract*—Vehicles on the road, where an Intelligent Transportation System (ITS) is built, can share a lot of traffic information and drive more safely and efficiently through data sharing. Since incorrect information misleads vehicles and causes confusion in traffic, a vehicular trust model is needed to check the message's trustworthiness while considering the vehicle's properties and protecting its privacy. In this paper, we proposed a two layered blockchain-based reputation system, which consists of a local one-day message blockchain and a global vehicle reputation blockchain. It can administrate the reputation score securely and preserve the vehicle's partial privacy. The proposed model efficiently manages local traffic information through the local one-day blockchain, reducing the memory overhead of vehicles. As the vehicle's actual identity and activities in other areas are hidden by using one-time public keys, partial privacy of the vehicle is preserved. According to the activity of the vehicle, the vehicle's reputation score is updated and stored permanently in the global reputation blockchain. We also suggested the location-based practical byzantine fault tolerance (LPBFT), a new consensus algorithm for fast block generating. The LPBFT lowers message propagation time through location-based primary node selection and is about 1.4 times faster than existing PBFT. The simulation results show the efficiency and the feasibility of LPBFT and our proposed protocol.**

***Index Terms*—Blockchain, security, vehicular reputation management.**

I. Introduction

With the development of autonomous vehicle technology, a vehicle's sensing abilities and communication capabilities have been rapidly improving. Based on these technologies, vehicles are able to generate and collect traffic information, and actively share it with other vehicles and RSUs (Road Side Units) in intelligent transportation systems (ITS) [1]. Sharing data between vehicles in traffic networks makes it possible to immediately respond to traffic accidents and establishes a safe and efficient traffic management system.

Manuscript received January 28, 2021; revised September 8, 2021; accepted November 15, 2021. Date of publication November 30, 2021; date of current version February 14, 2022. This work was supported in part by the MSIT (Ministry of Science, and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2018-0-01417) supervised by the IITP (Institute for Information and Communications Technology Planning & Evaluation, and in part by the National Research Foundation of Korea through the Korea Government under Grant 2018R1A2B6006903. The review of this article was coordinated by Prof. Jian Weng. *(Corresponding author: Seung-Hyun Seo.)*

Soojin Lee is with the Department of Electrical Engineering, Graduate School of Hanyang University, Seongdong-gu, Seoul 04763, South Korea (e-mail: tssn195@hanyang.ac.kr).

Seung-Hyun Seo is with the Division of Electrical Engineering, ERICA Campus, Hanyang University, Ansan, Gyeonggi-do 15588, South Korea (e-mail: seosh77@hanyang.ac.kr).

Digital Object Identifier 10.1109/TVT.2021.3131388

However, if there is a malicious vehicle or RSU in the data sharing process, it could cause confusion that leads to dangerous situations such as car accidents, by transmitting a false message that does not match the actual traffic situation such as a reported accident where none happened. This can affect navigation systems, traffic police, and actually cause traffic accidents [2].

So far, to deal with these types of situations, several vehicle trust management systems have been leveraged to enhance the traffic data trustworthiness and improve the vehicle's ability to judge message reliability [3]–[10]. In these previous works, A specific vehicle's reputation is determined based on the neighboring vehicles' opinions as well as its own past activities. This reputation is used to measure its trustworthiness in future network interactions. Through this system, vehicles can determine a message's trustworthiness. If a central authority manages the vehicles' reputation information, it may be a target for hackers or have too much management overhead. To overcome the limitations, reputation information can be stored distributively through multiple traffic servers such as an RSU [7]–[11]. However, there are still challenges in synchronization, reliability and trust service due to the possibility of RSU malfunction or intrusion.

Recently, blockchain technology [12] has been gaining popularity and interest for its application in vehicular trust models as a way to overcome the issues discussed above. In the blockchain network, participants share and store ledgers that keep the traffic information and vehicle's reputation scores. Blockchain technology guarantees data integrity with cryptographic techniques. Many studies have been conducted by utilizing this blockchain system for vehicular trust and reputation management [2], [13]–[22]. However, previous studies have not taken into account the characteristics of the vehicle network environment or have problems protecting the privacy of vehicles and maintaining the blockchain ledger efficiently.

In this paper, in order to guarantee efficient ledger management and address the privacy issues, we proposed a two-layered vehicle reputation blockchain system consisting of a local one-day message blockchain and a global vehicle reputation blockchain. The proposed model reduces the burden on the vehicle by temporarily storing the traffic information of the region through the local one-day message blockchain. It also provides partial privacy to the vehicle by managing the reputation of all the vehicles with the global vehicle reputation blockchain. Because vehicles only use the traffic information from the roads being traveled on, there is no need to know any information outside the region the vehicles are in. The local

one-day message blockchain is a temporary one which stores local traffic information created for one-day and is destroyed the next day. This allows the vehicle to store a lightweight blockchain, reducing memory overhead and directly accessing local traffic information. Since the reputation information of the vehicle must be recorded permanently, it is managed long-term through the global vehicle reputation blockchain where the RSU is a full node. Whenever a vehicle joins a new local one-day message blockchain network, it generates a temporary public key from the long-term public key and uses it as a pseudonym. This effectively hides the actual identity of the vehicle but shares the accumulated reputation score. Therefore, they know the temporary identity of the neighboring vehicles, but do not know their actual identities.

In addition, due to the ever-changing and dynamic vehicular environment, local one-day message blockchains require fast consensus algorithms such as practical byzantine fault tolerance (PBFT) [23]. To this end, in order to select a more efficient and safe mining node, we suggest the location-based practical byzantine fault tolerance (LPBFT) algorithm, which improves the existing PBFT for the local one-day message blockchain. In LPBFT, the RSU located closest to the event location recorded in the message is designated as the primary node so that it can check the trustworthiness of the message by sensing traffic data and can immediately begin the first step of the consensus algorithm. We simulated the LPBFT and our local one-day message blockchain network by using the Omnet++ simulator and Python3 Idle. As a result, we showed the effectiveness of our proposed system.

This paper provides the following contributions through the two-layered vehicle reputation blockchain system.

- We have proposed a two-layered vehicle reputation blockchain system, which operates a local one-day message blockchain and global vehicle reputation blockchain, for efficient traffic data sharing while considering the vehicle's properties such as mobility and limits of memory storage and computing power.
- We preserved the vehicle's partial privacy by using the one-time public key generated based on the actual identity of the vehicle in the local one-day message blockchain. Although vehicles have a knowledge of the nearby vehicle's reputation level and one-time account, the overall activity of the vehicle would not be revealed to other vehicles.
- We suggested the new consensus algorithm, LPBFT for effective operation of local one-day message blockchains in vehicular networks.

The rest of this paper is organized as follows: we discuss related works for trust model in vehicular networks in Section II. We propose our two layered blockchain-based reputation system including the adversary model and security requirements in Section III. We propose LPBFT algorithm and present a local one-day message blockchain and a global vehicle reputation blockchain protocol in Section IV. In Section V, we analyze the security of our proposed model. In Section VI, we provide the simulation results and performance analysis. We describe conclusions in Section VII.

II. Related Works

A. Trust Model in Vehicular Networks

Previous studies [3]–[6], [24] proposed system models and methods for vehicles to directly collect opinions and local information from other vehicles and then calculate the reputation of the received messages in vehicular networks. Engoulou *et al.* [3] presented a set of local parameters that the vehicle uses to calculate other vehicle's local reputation. By adding the local reputation and indirect reputation which the surrounding vehicles compute, the vehicle can find the final reputation score of other vehicles. Li *et al.* [4] proposed an attack-resistant trust management scheme to detect malicious attacks and measure concurrently the reliability of the received messages. However, in [3] and [4], since the vehicle directly stores the reputation scores of other vehicles in its internal memory, all stored information can be lost in the event of a vehicle breakdown or security attack. Chen *et al.* [5] introduced a trust-based message propagation scheme, which applied an improved cluster-based data routing mechanism to gather opinions from the surrounding vehicles. The gathered opinions determine whether the information is reliable or not. However, their scheme requires additional methods for electing an honest cluster leader and hiding the linking of the vehicle's identity and its opinions. Chikhaoui *et al.* [6] used time and location data to determine the accuracy of the received message, and calculate the reputation based on the number of messages with the same content received from other vehicles and their reputations. To preserve the vehicle's privacy, it applied ticket-based authentication [25]. Magaia *et al.* [24] proposed a novel reputation framework for information-centric vehicular applications using machine learning such as Bayesian learning and K-Means clustering. Unfortunately, since each vehicle has its own reputation table in [6], [24], it is difficult to consider all past behaviors of a specific vehicle, so the accuracy and reliability of the reputation information are low.

Some methods [7]–[11] managed and stored a vehicle's information with the help of a third party such as an RSU. Constantino *et al.* [7] designed the context aware reputation systems to detect and respond to denial-of-service attacks. However, they did not propose a specific vehicle reputation calculation method. In order to verify vehicles' message reliability, Oluoch [8] proposed a distributed reputation model that applied a combination of the vehicle's reputation information and the opinions of surrounding vehicles. The information is obtained when the vehicle requests service from the RSU. However, the proposed scheme has difficulty synchronizing reputation information between RSUs in real time. Huang *et al.* [9] designed a distributed reputation management system. By using multi-weighted subjective logic, they tried to update the reputation information more accurately. However, [9] is highly dependent on the Vehicular Edge Computing server. Therefore if there are a number of vehicles in the area, the RSU may not be able to provide high quality service. Tangade *et al.* [11] suggested a trust management scheme based on hybrid cryptography (TMHC). An agent trusted authority (ATA) computes a vehicle's trust value based

on the RSU's assessment of the vehicle. As the ATA manages all of the vehicle's trust value, there is a possibility of a security attack targeting the ATA. Muhammad *et al.* [10] proposed trustVote, a crowdsourcing-based vehicle reputation system. The model hid the voting score by using a homomorphic encryption algorithm, but it requires a high computation overhead for vehicles.

B. Blockchain Based Trust Model in Vehicular Network

In order to solve the limitations of the previous trust models described in the above section, studies about vehicular trust models which store vehicle reputation information using the blockchain for transparency and security are being actively conducted [2], [13]–[19], [22]. As the blockchain continues to grow, the participants, which act as full nodes, must have enough memory storage to keep up with the growth and they need more and more computational power for cryptographic calculations for mining. To guarantee complete decentralization and the security of the blockchain network, it is essential to determine which participants will store and manage ledgers in the decentralized blockchain-based trust model. From previous studies, we figured out that blockchain-based trust models can be divided into three categories according to this ledger management: 1) Vehicle [2], [13], [14], 2) RSU [15]–[17], [22], and 3) both Vehicle and RSU [18].

In the first case, the vehicle stores all historical blocks as a full node, so vehicles can directly access the vehicle's reputation and traffic messages in the blockchain network. To the best of our knowledge, Yang *et al.* [2] first applied blockchain technology to a vehicle reputation system. The vehicles share the blockchain ledger and store the voting records of the message. Lu *et al.* [13] proposed a blockchain-based anonymous reputation system (BARS) which used direct historical interactions and indirect opinions about the vehicle as evidence of the vehicle's reputation. In the system, when the real identity of the public key owner is revealed, the public key is revoked to protect the privacy of the vehicle. The activities of the CA, which has the responsibility of updating the certificate and public keys of vehicles, are recorded openly in multiple blockchains, so it can be monitored. Shrestha *et al.* [14] designed a blockchain system with independent local blockchains in each country for a scalable and efficient distribution of blocks. However, in their system, there is a privacy issue where the activity history of the vehicle is exposed to the public. In [2], [13], [14], since vehicles run the proof of work (PoW) to mine blocks, they have to consume a lot of power. So, the most advanced vehicles may be selected as miners frequently due to the differences in computational power between vehicles, which means the whole system is not fully decentralized. In addition, vehicular network participants must initially download massive blockchain data, which is time-consuming and requires more memory as the blockchain ledgers grow over time.

In the second case, the RSU manages a blockchain ledger and stores the vehicle's reputation score while the vehicles send requests to obtain traffic information or vehicle reputation scores. Kang *et al.* [15] proposed a reputation-based data sharing scheme using a vehicle's local opinions for data credibility. RSUs create the blocks aggregating the vehicle's sensing data stored in edge nodes. Yang *et al.* [16] proposed a blockchain-based decentralized trust management system, which applied a new consensus algorithm combining PoW and proof of stake (PoS) to make RSUs to include as many transactions as possible in a block, so that real-time traffic information is shared quickly in the network. Iqbal *et al.* [22] presented vehicular fog network architecture for tasks offloading. RSUs assign task to neighboring vehicles based on their social reputation score stored in semi-private consortium blockchains. In [15], [16], [22], the RSUs are miners in a fixed position, so a ledger synchronization is more efficient than when the vehicle is the miner. However, the vehicle must make a request to the RSU to acquire the block information. That is why it is difficult to see a completely decentralized system and to detect malicious RSUs. In order to minimize the chance of a malicious RSU's attack, Kang *et al.* [17] calculated the reputation of a RSU based on the past communications and opinions of vehicles. But, they did not consider how to verify the reliability of data sent by the vehicle.

In order to overcome the disadvantages of the first and second cases, Kandah *et al.* [18] suggested a Global Trustworthy System with a multi-layer blockchain structure. The platoon blockchain in which vehicles participate as nodes stores the trust score of the vehicle, and the global blockchain periodically stores the platoon blockchain. The trust-bidding system is used as a consensus algorithm considering the vehicle's computational power and mobility. However, the process for estimating the score of the vehicle has not been clearly presented. In addition, there is a possibility that a false message may be included in the block because a miner vehicle cannot check the reliability of the messages reporting an accident which it has not witnessed.

In addition to ledger management, we must also consider the privacy issues of vehicles [26]. Because of the transparency of the blockchain, the reputation scores and activity history of all vehicles are exposed to all participating entities. Luet *et al.* [13] attempted to provide vehicle anonymity by periodically updating the public key, through the Certificate Authority (CA). However, this is an unrealistic method due to the overhead caused by the CA issuing many public keys. Li *et al.* [19] proposed a Creditcoin, which applied the threshold ring signature to blind the vehicle's message voting history. The proposed system incentivizes the vehicles according to their contribution in the network to lead the vehicle's active participation. Unfortunately, it is complicated and time-consuming for vehicles to verify this signature. Moreover, since it provides full anonymity, the reputation information of neighboring vehicles is not known. However, the vehicle needs to know the reputation level of the surrounding vehicles because the reputation information helps the vehicle decide whether to trust the message. Therefore, it is necessary for the vehicle to know the reputation value of the neighboring vehicles and at the same time preserve the privacy of the actual identity of all the vehicles. Related studies of blockchain-based vehicle trust and reputation systems are outlined in Table I .

TABLE I
A COMPARATIVE STUDY OF BLOCKCHAIN-BASED VEHICLE TRUST SYSTEMS

Paper	Ref. No.	Description	Blockchain Type	Miner	Consensus	Privacy
Yang	[2]	proposed a new blockchain-based reputation system for data credibility assessment	Public	Vehicle	PoW	·
Lu	[13]	proposed a blockchain-based anonymous reputation system	Public	Vehicle	PoW	✓
Shrestha	[14]	introduced a new type of blockchain, which is named local blockchain	Public (based on location)	Vehicle	PoW	·
Kang	[15]	proposed a reputation-based data sharing scheme in vehicular network	Consortium	RSU	PoW	✓
Yang	[16]	proposed a new decentralized trust management scheme for vehicular network	Private	RSU	a joint PoW and PoS	·
Kang	[17]	proposed an enhanced DPoS scheme with a two-stage system model for vehicular blockchain	Private	RSU	an enhanced DPoS	·
Iqbal	[22]	proposed a blockchain-based reputation management framework for task offloading using fog vehicles	Semi-consortium	RSU	PoET	·
Kandah	[18]	suggested a multi-tier blockchain-based trust management framework for secure vehicular network	Private	Vehicle /RSU	trust-bidding \PoW	·
Li	[19]	proposed Creditcoin, which is a privacy-preserving blockchain-based incentive system using threshold ring signature	Public	RSU or official public vehicles	BFT	✓

TABLE II
LIST OF NOTATIONS

v_i	Unique identifier of a $vehicle_i$
v_i^{OT}	An one-time account of a $vehicle_i$ in global vehicle reputation blockchain
R_j	Unique identifier of a RSU_j
q	A primer number
l	A length of blockchain account
h_0	A hash function : $\{0,1\}^* \rightarrow \mathbb{Z}_q\ \mathbb{F}_q$
h_1	A hash function : $\{0,1\}^* \rightarrow \{0,1\}^l$
$E/\mathbb{F}_q$	The selected eliptic curve over the field $\mathbb{F}_q$ $y^2 = x^3 + ax + b \pmod q, a, b, x, y \in \mathbb{F}_q$
P	Point generator of an additive cyclic group $\mathbb{G}_q$
sk_i^v	A private key of $vehicle_i$ in global vehicle reputation blockchain
pk_i^v	A public key of $vehicle_i$ in global vehicle reputation blockchain
sk_j^r	A private key of RSU_j
pk_j^r	A public key of RSU_j
O_{sk}^i	A private key of $vehicle_i$ in local one-day message blockchain
O_{pk}^i	A full public key of $vehicle_i$ in local one-day message blockchain
$ReportingM$	An accident reporting message packet
$VotingM$	A voting message packet
N	The number of $votingM$ which the $vehice_i$ should collect
T	A timestamp when the transaction or message is built
msg	Traffic information including event location
$coinID_{v_i}$	Unique identifier of $vehicle_i$'s voting coin in local one-day message blockchain
$Location$	A location information of a $vehicle_i$
tx_{input}^i	An input of message transaction including a coin ID and signature of $vehicle_i$
tx_{vote}^r	An input of message transaction including a coin ID and signature of $recip_r$
tx_{output}^j	An output of message transaction which the recipient is RSU_j

III. TWO LAYERED VEHICLE REPUTATION BLOCKCHAIN SYSTEM

In this section, we explain our proposed system model and describe how vehicles share reporting and voting messages in our two layered reputation blockchain system. We assume that all vehicles have access to the location information and public key list of RSUs located in their region. The list of notations is shown in Table II.

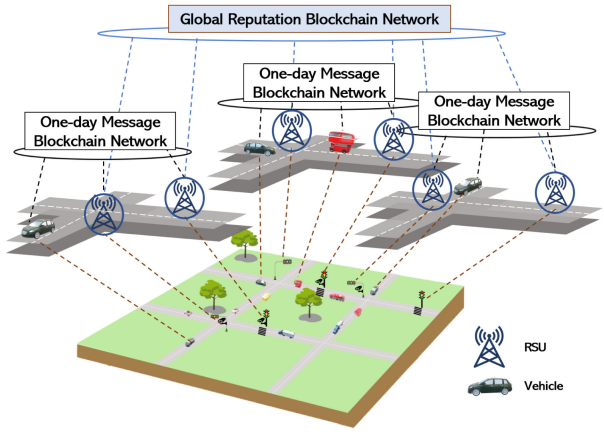


Fig. 1. Overall model.

A. System Model

We introduce the participants and blockchains of the two-layered vehicle reputation system model shown in Fig. 1 such as local one-day message blockchain and global vehicle reputation blockchain. Then, we discuss the adversary model and security requirements in proposed system.

1) Participants: The main participants of the proposed model are vehicle, RSU, and certificate authority.

- *Vehicle*: Vehicles store a local blockchain ledger as a full node, so they can directly read local traffic information. The role of the vehicle in our system is divided into message sender and message receiver in the local one-day message blockchain. When a message sender observes an accident or collects traffic information, he creates a message and distributes it to nearby vehicles and RSUs to share the

information. The recipient vehicles of the message can vote on whether they agree it is trustworthy or not. Both message senders and receivers can increase their reputation scores through their activities in the vehicle network.

- *RSU*: A RSU is located across the road and has an obligation to mine and share blocks on both global vehicle reputation blockchains and local one-day message blockchains. The RSU is assumed to be an almost fully trusted party in the proposed system. Even if the RSU malfunctions due to security attacks, they are quickly detected by the nearby RSU and do not significantly affect the whole operation of the system. The RSU's location is fixed and it has high computing power. The RSUs validate transactions, which are made by the vehicle in the local one-day message blockchain, and blocks received from other RSUs. In addition, when the vehicle first participates in the blockchain, the RSU verifies its identity and allows the vehicle to participate in the blockchain, so the RSU knows the mapping of identities and one-time accounts that the vehicle uses temporarily in the region.
- *Certificate Authority (CA)*: The certificate authority issues digital certificates for a private key-public key pair, which is generated by a driver, when he first registers the vehicle. The certificates guarantee that the driver has ownership of the key. The generated public key is used as an account in the global vehicle reputation blockchain.

2) Blockchains: Two-layered blockchain system for vehicle reputation is operated in a hierarchical structure which we are calling *Local One-day Message Blockchain* and *Global Vehicle Reputation Blockchain*.

- *Local One-day Message Blockchain*: As its name implies, local one-day message blockchain stores transactions that occur during the day locally. Local traffic events such as traffic accident control and construction can last for several hours, so vehicles participating in the area after the event should also be able to get that traffic information for efficient driving. Therefore, vehicles and RSUs store and share local traffic information in the short term through the local one-day message blockchain. The RSUs and vehicles in the region participate as blockchain nodes. The region of the local one-day blockchain means an area where the generated traffic information has a major influence on vehicles in the area. When a vehicle moves to another region, traffic information of the previous area does not affect its activities in its current region. The vehicle does not need traffic information from other cities or faraway areas of the same city when the vehicle is driving. Therefore, vehicles do not need to maintain one-day blockchain ledgers in different regions.

  Due to the public nature of the blockchain system, even though participants use pseudonyms, it is possible for others to learn the ID of the real users or their personal information by tracing the flow of transactions recorded in the ledger. Particularly in the vehicle networks, the user's driving region and route are fairly constant. So if the same one-time public key is used as a pseudonym in all local one-day message blockchains, even though the vehicle's regional activity details are stored in different local one-day message blockchain ledgers, the activity pattern of the vehicle can be revealed by another vehicle. In order to prevent the privacy leakage, in the proposed model, the vehicle uses a different public key for each region and day. When a vehicle tries to participate in the network, the vehicle initially generates a one-time public key to use as a pseudonym in the local one-day message blockchain. The vehicle uses the long-term public key to generate the self-signing certificate to sign the one-time public key. The RSU can then approve the one-time public key in the local one-day message blockchain.

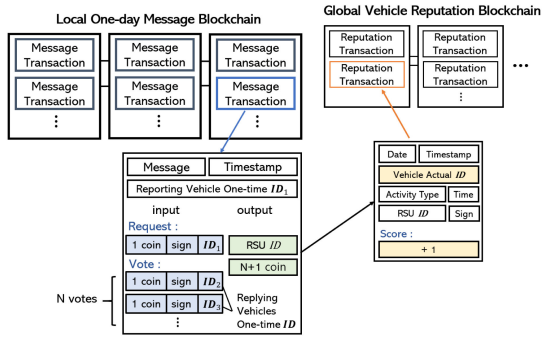


Fig. 2. A structure of transactions.

The vehicle pays coins to do activities in the local one-day message blockchain. Every coin has a unique ID and all remittance history is recorded in the local one-day message blockchain. The coin in our model is used for requests and voting rights so we assume that every coin has the same value. Each coin has a unique identifier, so that every user's coins can be easily distinguished. To use a coin, the user must sign the coin to prevent others from stealing the coin. A transaction in a local one-day message blockchain includes a reporting message and a list of voting coins showing vehicles' agreements with the message. The input of each transaction consists of a request coin, voting coins, voted vehicle's signatures, and uses the one-time public key as a temporary vehicle ID. The total spent coins are transferred to a nearby RSU, which records the total coins spent in the output of the transaction as illustrated in Fig. 2. When the transaction passes verification and is recorded in the blockchain ledger, the coins in the transaction has been spent.

The local one-day message blockchain newly creates a genesis block at a set time every day and deletes the previously recorded blockchain data. The block miner is the RSU, not the vehicle, in order to prevent unstable communication problems caused by the vehicle's non-coordination and absence. For a quick and efficient consensus, we propose a new concept of consensus algorithm which is a location based practical byzantine fault tolerance (LPBFT). Both PBFT and LPBFT require the process of propagating the transaction to other RSUs in order for the primary node to reach consensus in the first pre-prepare stage. Before starting this step, it is necessary to determine the primary

node in advance. The PBFT executes the view change process to select a new primary node, and for this, transaction transfer between participating nodes is required. However, in our proposed LPBFT, the RSU closest to the location where the transaction occurred automatically becomes the primary node, so the process to select the primary node is eliminated. In addition, the consensus process starts only when the primary RSU receives a transaction for consensus from the vehicle. In order to proceed with the consensus process between consensus nodes, the primary node must first know the transaction information. In PBFT, the primary node, which initiates and leads the consensus process of transactions between RSUs, is randomly selected. Accordingly, if the randomly selected primary RSU is located far from where certain traffic information occurred, it may not be able to verify that the transaction of that information is accurate. On the other hand, the RSU located closest to the accident location can verify the authenticity of the accident by using its sensors such as cameras, an accelerometer, and a lazer scanner [27]. The pseudo code of LPBFT is shown in Algorithm 1. In Algorithm 1, $h()$ is a cryptographic hash function that receives a message transaction as an input and outputs a hash value of a fixed length.

- *Global Vehicle Reputation Blockchain*: Global vehicle reputation blockchain is a private blockchain in which RSUs located in different regions participate as nodes. The RSU periodically updates 9 the reputation score of the vehicles according to the activities of each vehicle recorded in the local one-day message blockchain as shown in Fig. 2. The RSU sends coins to the vehicle when it creates a new one-time public key. The number of coins used for the vehicle's activities is determined in proportion to the vehicle's reputation. Thus, the lower the vehicle's reputation is, the lower the number of coins that can be used. Therefore, the updated reputation score of the vehicle affects the amount of vehicle activity when participating in the next local one-day message blockchain. While the RSU can read the reputation scores of all the vehicles, the vehicle cannot access the reputation blockchain information. Therefore, it should request the reputation information from the RSU if it is needed. The vehicle may indirectly know the reputation information of the surrounding vehicles through the RSU, but the RSU hides the actual identifier of the vehicle when it responds to the vehicle's request. As a result, the connection between the current vehicle's temporary address and reputation score is revealed, but the relationship between the vehicle's identity and reputation score is hidden. The global vehicle reputation blockchain is a private blockchain that only RSU participates in, so the PBFT, which is suitable for private blockchains [28], is used as a consensus algorithm of the global vehicle reputation blockchain.

Algorithm 1: Pseudo Code of LPBFT.

Input: Message Transaction mt, Hash Function $h()$, Timestamp T.

Upon reception of REQUEST at RSU_p **do**
if $p = argmin_j(dis(RSU_j, accidentlocation))$ **then**
  if *verify*$(mt) ==$ *true && *$h(mt)$ *is valid* **then**
    PRE-PREPARE(mt)
  else
    error("Validation Failed")
else
  error("It's not a nearest RSU")

Function PRE-PREPARE(mt) **at** RSU_p **do**
  multicast $<$ PRE-PREPARE, $mt, h(mt), T >$ to other RSU

Upon reception of PRE-PREPARE at RSU_j **do**
  multicast <PREPARE, $mt, h(mt), T >$ to other RSU

Upon reception of 2*f* PREPARE at RSU_j **do**
if *verify*$(mt) ==$ *true && *$h(mt)$ *is valid* **then**
  multicast <COMMIT, $mt, h(mt), T >$ to other RSU
else
  error("Validation Failed")

Upon reception of 2*f* COMMIT at RSU_j **do**
  add mt in blockchain ledger
  broadcast to nearby $vehicle_i$

B. System Design

Setup: Each RSU and vehicle generates a pair of private and public key and requests a certificate from the *CA*. The *CA* registers each RSU and vehicle by including a list of participants.

1) RSU Registration: An RSU_j $(j = 1, \ldots, M)$ chooses a uniformly distributed random secret value $sk_j^r \in \mathbb{Z}_q^*$ and computes $pk_j^r = sk_j^r P$. The RSU_j sends pk_j^r to the *CA* for registration. The *CA* issues a certificate to the RSU_j. $R_j = h_0(pk_j^r)$ is used as a unique ID of the RSU_j.

2) Vehicle Registration: A $vehicle_i$ $(i = 1, \ldots, m)$, which is to be newly registered, chooses an uniformly distributed random secret value $sk_i^v \in \mathbb{Z}_q^*$ and computes $pk_i^v = sk_i^v P$. The $vehicle_i$ transfers pk_i^v to the *CA* and gets a certificate for the key pair (sk_i^v, pk_i^v). $v_i = h_0(pk_i^v)$ is used for the $vehicle_i$'s unique identifier. v_i and pk_i^v are transferred to all RSUs.

Joining: To join the one-day message blockchain, a vehicle must generate an one-time private-public key pair and create a self-signed certificate for the one-time public key, as follows:

1) A $vehicle_i$ initially chooses a uniformly distributed random value $O_{sk}^i \in \mathbb{Z}_q^*$ as a one-time private key and calculates a one-time public key $O_{pk}^i = O_{sk}^i P$. $v_i^{OT} = h_0(O_{pk}^i)$ is a one-time account of $vehicle_i$ in the local one-day message blockchain.
2) The $vehicle_i$ creates a self-signed certificate $c_i = (\gamma_i, \delta_i)$ for the one-time public key O_{pk}^i by using sk_i^v.
   - It chooses a uniformly distributed random value k_i $(1 \leqq k_i \leqq q-1)$ and computes $\gamma_i = k_i P$ and $\delta_i = (h_1(Location || O_{pk}^i || T) + sk_i^v \gamma_i) k_i^{-1} \pmod q$
3) The $vehicle_i$ creates the following packet and sends it to the surrounding RSU_j:

$$\{Location, T, O_{pk}^i, c_i, v_i\}$$

4) Once RSU_j receives the packet, it checks if $Location$ matches the vehicle's real-world location.
   - The RSU_j computes $e_i = h_1(Location||O^i_{pk}||T) \cdot {\delta_i}^{-1} \pmod q$ and $t_i = \gamma_i \cdot {\delta_i}^{-1}$.
   - If $e_i P + t_i pk^v_i = \gamma_i$, the verification of c_i is successful.
5) Then the RSU_j sends a local one-day message blockchain ledger and a list of participating vehicles to the $vehicle_i$. The RSU_j checks the $vehicle_i$'s reputation score recorded on the global blockchain, and sends the coins to $vehicle_i$ based on $vehicle_i$'s reputation score. The RSU transfers the same amount of coins if the vehicle's reputation score is 0.5 or higher, otherwise the lower amount of coins will be paid proportionally to the reputation score. The coin is used when the $vehicle_i$ generates a message transaction or vote for received messages.

Deploying Transactions: The reporting vehicle broadcasts a traffic message to more than N nearby vehicles, where N should be more than half of the surrounding vehicles. Neighboring vehicles, who agree with the message, send a vote to the reporting vehicle. After collecting N votes, the reporting vehicle creates a message transaction and sends it to the surrounding RSU.

1) $Request$: The $vehicle_i$, which witnesses an accident, generates $ReportingM$ as follows:
   - It chooses an RSU_j, which is located near the $vehicle_i$. The transaction output tx^j_{output} is $\{N+1, R_j\}$, where $N+1$ is the total coin value used for requesting and voting.
   - The $vehicle_i$ selects a $coinID_{v_i}$ among his unspent coins.
   - The $vehicle_i$ chooses a uniformly distributed random value d_i $(1 \leqq d_i \leqq q-1)$. Then it computes $\mu_i = d_i P$ and $\upsilon_i = (h_1(msg||coinID_{v_i}||T||v^{OT}_i||tx^j_{output}) + O^i_{sk}\mu_i){d_i}^{-1} \pmod q$ to generate a signature on $ReportingM$, $s_i = \{\mu_i, \upsilon_i\}$.
   - The transaction input tx^i_{input} is $\{coinID_{v_i}, v^{OT}_i, s_i\}$
   - A $ReportingM$ is
   $$\{msg, v^{OT}_i, coinID_{v_i}, tx^i_{input}, tx^j_{output}, T\}.$$
   - The $vehicle_i$ broadcasts this $ReportingM$ to more than N neighboring vehicles.
2) $Verification\ of\ ReportingM$: Each recipient vehicle $recip_r$ $(r = 1, \ldots, m_0, r \neq i)$ validates the $ReportingM$.
   - $recip_r$ computes $\alpha_i = h_1(msg||coinID_{v_i}||T||v^{OT}_i||tx^j_{output}) \cdot {\upsilon_i}^{-1} \pmod q$, and $\beta_i = \mu_i \cdot {\upsilon_i}^{-1}$.
   - If $\alpha_i P + \beta_i O^i_{pk}$ is equal to μ_i, the s_i is valid.
3) $Reply$: If a $recip_r$ agrees with the $vehicle_i$'s $ReportingM$, it responds by sending a $VotingM$. To prevent attackers from stealing the vehicle's $votingM$, each $recip_r$ generates a signature S_r including tx^i_{input}.
   - Each $recip_r$ chooses a uniformly distributed random value b_r $(1 \leqq b_r \leqq q-1)$. To generate a signature $S_r = (\epsilon_r, \theta_r)$, it computes $\epsilon_r = b_r P$ and $\theta_r = (h_1(msg||coinID_{v_r}||T||v^{OT}_r||tx^i_{input}||tx^j_{output}) + O^r_{sk}\epsilon_r){b_r}^{-1} \pmod q$ Then, it makes $tx^r_{vote} = \{coinID_{v_r}, v^{OT}_r, S_r\}$.
   - The created $VotingM$ is $\{msg, v^{OT}_r, tx^r_{vote}, T\}$
4) $Announcement$: When N $VotingM$s are collected, the $vehicle_i$ constructs a message transaction including N $VotingM$s, then broadcasts it to RSU_j. The message transaction is as follow:
   $$\{msg, v^{OT}_i, tx^i_{input}, tx^{r_1}_{vote}, tx^{r_2}_{vote}, \ldots, tx^{r_N}_{vote}, tx^j_{output}, T\},$$
   where $tx^{r_n}_{vote}$ $(n = 1, \ldots, N)$ represents the n-th $recip_r$'s transaction input.

Mining: For miner selection and block deployment, we used the following proposed consensus algorithm, *Location based Practical Byzantine Fault Tolerance (LPBFT).*

1) The RSU_p, which is located closest to the accident location recorded in the message transactions, becomes the primary RSU to propagate the transaction.
2) $Verification\ for\ Reporting\ Vehicle's\ Signature$: The RSU_p verifies the signature s_i of $vehicle_i$.
   - The RSU_p computes $\alpha_r = h_1(msg||coinID_{v_i}||T||v^{OT}_i||tx^j_{output}) \cdot {\upsilon_i}^{-1} \pmod q$ and $\beta_i = \mu_i \cdot {\upsilon_i}^{-1}$.
   - If $\alpha_i P + \beta_i O^i_{pk} = \mu_i$, the $vehicle_i$'s signature s_i is validated.
3) $Verification\ for\ Replying\ Vehicle's\ Signature$: The RSU_p check if all $recip_r$'s votes in the message transaction are valid.
   - For each $tx^{r_n}_{vote}$, the RSU_p computes $\sigma_{r_n} = h_1(msg||coinID_{v_{r_n}}||T||v^{OT}_{r_n}||tx^i_{input}||tx^j_{output}) \cdot \theta^{-1}_{r_n} \pmod q$ and $\tau_{r_n} = \epsilon_{r_n} \cdot {\theta_{r_n}}^{-1}$.
   - If $\sigma_{r_n} P + \tau_i O^{r_n}_{pk} = \epsilon_{r_n}$ for all n, the verification is successful.
4) $Pre-prepare$: The primary RSU_p broadcasts $< PRE-PREPARE, mt, h_1(mt), T >$, in which the mt is a message transaction and the $h_1(mt)$ is used for message integrity, to all the remaining RSUs. Each RSU_j, who receives $PRE-PREPARE$, verifies whether the signature of the message transaction is valid or not by performing the above calculation from step 2) and 3) like the primary RSU_p.
5) $Prepare$: After the verification, each RSU_j broadcasts $< PREPARE, mt, h_1(mt), T >$ to all other RSUs. Once they collect $\frac{2(M-1)}{3}$ $PREPARE$, the RSU_j is ready for the next step.
6) $Commit$: Each RSU_j broadcasts $< COMMIT, mt, h_1(mt), T >$ to all other RSUs. Once they collect $\frac{2(M-1)}{3}$ $COMMIT$, which contains the same message that RSU_j received in the $prepare$ step, they add the transactions in the local one-day message blockchain ledger.
7) Each RSU_j broadcasts the newly stored transactions and block to the nearby vehicles. Vehicles, who receive the block, also share it with other neighboring vehicles.

Leaving: When the vehicle leaves the local one-day blockchain area, the RSU updates the list of participants.

1) Once the $vehicle_i$ leaves the location, the RSU_j on the regional boundary can detect the departing $vehicle_i$.

2) The RSU_j announces that the $vehicle_i$ has left the local one-day message blockchain network.
3) The $vehicle_i$ can delete all memory of the previous one-day message blockchain ledger and has to execute the $Joining$ phase for the next area.

Updating Reputation Score: According to the vehicle's activity history in the local one-day message blockchain, an RSU updates the vehicle's reputation using the beta reputation function [29] and records the updated score in the global vehicle reputation blockchain. Each vehicle has different probability of honest coin use H. A beta distribution $beta(H|pos, neg)$, where $0 \leq H \leq 1$ and the parameters $pos, neg > 0$, can be defined using gamma function as:

$$beta(H|pos, neg) = \frac{\Gamma(pos + neg)}{\Gamma(pos)\Gamma(neg)} H^{pos-1}(1-H)^{neg-1}$$

The pos and neg are the count of negative and positive behavior respectively. The probability expectation value of the beta distribution function is $\frac{pos}{pos+neg}$, which is used as reputation score. The range of reputation score is [0, 1].

1) Each RSU_j counts the number of coins stored in the local one-day message sc and the number of coins not stored in the local one-day message blockchain sc' among the coins used by the $vehicle_i$.
2) Each RSU_j calculates reputation score using the beta reputation function as follows:

$$Score = \frac{pos}{pos + neg}$$

The pos and neg are equal to $sc + sc_{old} + 1$, and $sc' + sc'_{old} + 1$ respectively, where sc_{old} and sc'_{old} are accumulated counts of coins stored in the local one-day message blockchain and the counts of coins not stored in the blockchain total the coins spent by the $vehicle_i$ respectively. Then it creates the reputation transactions rx to update $vehicle_i$'s reputation score as follows:

$$rx = \{R_j, AC_{id}, Date, T_{AC}, v_i, Score, T\},$$

where AC_{id} is a type of vehicle activity including transaction creation, voting, malicious behavior, etc, $Date$ and T_{AC} denote date and time of the day the vehicle did the activity AC_{id}, $Score$ is the newly updated reputation score.
3) Before propagating the created transaction, rx, the RSU_j generates signature r_j=$\{\eta_j, \zeta_j\}$ on rx.
   - Each RSU_j chooses a uniformly distributed random value $f_j(1 \leqq r_j \leqq q-1)$. And then it computes $\eta_j = f_j P \pmod q$ and $\zeta_j \equiv (h_1(rx) + sk_j\eta_j){r_j}^{-1} \pmod q$.
4) Each RSU_j broadcasts the reputation transaction, rx and the signatures, r_j to other RSUs.
5) By using PBFT algorithm, the new block, which stores reputation transactions, is mined in the global vehicle reputation blockchain.
6) Every $vehicle_i$ can check its updated reputation score by asking a nearby RSU_j.

IV. Adversary Model and Security Analysis

In this section, we define adversary model and security requirements. In addition, we discuss how the proposed model works for the following security attack.

A. Adversary Model

In our proposed system, there are two types of adversaries: malicious vehicles and compromised RSUs [16]. According to PBFT's assumptions [23], when the total number of RSUs is M, we assume that no more than $\lfloor \frac{M-1}{3} \rfloor$ of RSUs are compromised in our system. Adversaries can attempt to reveal the overall network activity of the vehicle by mapping the activity history and actual identity of the vehicle stored in the blockchain. A malicious vehicle can report a fraudulent traffic message and try to distribute it to the surrounding vehicles to cause confusion in the traffic information system. It can also attempt to falsify the contents of a message transaction created by another vehicle or steal another vehicle's vote. Moreover, a malicious vehicle can create multiple virtual accounts for Sybil attacks as if there were more vehicles than there actually were on the road. A bunch of malicious vehicles may agree to forge traffic information and then create a valid transaction which has a fake message. A compromised RSU may try to deploy a false reputation transaction to deliberately manipulate a vehicle's reputation score in the global vehicle reputation blockchain network.

B. Security Requirements

The following security properties are required to support secure vehicle reputation networks for the two layered vehicle reputation blockchain system.

1) Preventing False Message Propagation: In the local one-day message blockchain network, malicious vehicles can generate false messages and propagate them to nearby vehicles. The system should be able to adjudge the reliability of traffic messages and prevent fake messages from being stored in blocks. To provide the requirement, the message voting process of nearby vehicles is necessary for evaluating the trustworthiness of shared messages.

2) Resistance Against Transaction Tampering: A malicious vehicle may try to alter the message or coins stored in the message transaction, which is sent by other vehicles, in the local one-day message blockchain network. In addition, a compromised RSU may manipulate a reputation transaction, so the system should be able to prevent the manipulated transaction from being included in the blockchain. The system must prevent any participating objects from tampering with the content of the transaction such as the voting coins, message, and reputation scores in both the local one-day message blockchain and global vehicle reputation blockchain.

3) Preserving Partial Privacy: It is crucial to protect the privacy of all vehicles' previous activities that they performed in the vehicular networks, such as message generation and voting. Even if a neighbor vehicle knows a vehicle's current behavior, the neighbor vehicle should not be able to map all the main

vehicle's previous movements in the local one-day message blockchains to its actual identity.

4) Preventing Sybil Attack: To prevent Sybil attack, the vehicle must be limited to using only one authorized account in the local one-day message blockchain. Moreover, the RSU must be able to verify that the vehicle does not use multiple identities at the same time.

5) Resistance Against a Malicious Vehicle Platoon: A malicious vehicle platoon may deliberately vote on the wrong traffic information and try to increase the reliability of the fake message in the local one-day message blockchain network. The system needs to be able to detect the fake message and prevent it from being mined to the block and propagated in the network.

6) Resistance Against Modifying or Robbing Coins: The malicious vehicle must not be able to modify invalid coins or reuse coins which have been spent before. Moreover, even if the vehicle knows the information of the other vehicle's coins, it should be impossible for the vehicle to use the coins.

7) Limiting the Activity of Malicious Vehicles: If an inappropriate behavior of a malicious vehicle such as dissemination of false information and forgery of transactions is discovered, the system must be able to limit the malicious vehicle's network activity in the local one-day message blockchain and impose a penalty.

C. Security Analysis

1) Preventing False Message Propagation: The malicious vehicle broadcasts a false message $ReportingM$, which is different from the actual traffic situation, to neighboring vehicles in the local one-day message blockchain. In order to make a valid transaction with its own false message, the malicious vehicle must receive a $VotingM$ from more than N surrounding vehicles. The malicious vehicle should collect votes from at least a majority of the vehicles in the vicinity. However, honest vehicles do not respond to the malicious vehicle's $ReportingM$. Even if a malicious vehicle accidentally collects enough $ReportingM$, the vehicle's reputation will be reduced if it is found to be a false message by the RSU.

2) Resistance Against Transaction Tampering: In the local one-day message blockchain, first, a malicious vehicle may try to forge the received $ReportingM$ or the transaction of another vehicle. If the attacker wants to tamper with the message, it must be able to forge other vehicles' signatures for the fake message. However, it is impossible to create valid signatures unless the attacker knows the other vehicle's one-time private key due to the difficulty of Elliptic Curve Discrete Logarithm Problem (ECDLP) [30]. Second, a compromised RSU also can attempt to manipulate a vehicle's reputation score in a reputation transaction rx in the global vehicle reputation blockchain. Nevertheless, it is quickly discovered by other RSUs who have shared the same local one-day message blockchain ledger. Once the compromised RSU's malicious behavior has been discovered, it is excluded from becoming a primary node in both the local one-day message blockchain and global vehicle reputation blockchain.

3) Preserving Partial Privacy: A malicious vehicle may attempt to map other vehicles' actual identities to their one-time identities to reveal the vehicles' entire activity histories. However, the one-time identity is generated differently for each region and date, so the full historical information of the vehicle is hidden from the attacker. Therefore, the proposed model preserves the vehicle's partial privacy.

4) Preventing Sybil Attack: A malicious vehicle may try to create a list of one-time accounts for vehicles that do not actually exist in the local one-day message blockchain network. However, when the malicious vehicle joins a local one-day message blockchain network, the RSU verifies the actual location of the vehicle and the one-time account for the vehicle by using the vehicle's actual identification (long-term public key). This information is shared with neighboring vehicles in the local area and other RSUs, which record and store the vehicle's one-time ID information. Therefore, if the malicious vehicle uses an unverified account, it is easily detected by the RSUs and neighboring vehicles.

5) Resistance Against a Malicious Vehicle Platoon: There is a possibility that some malicious vehicles form a group to create a transaction with false information in the local one-day message blockchain. In this case, during the transaction verification process, a neighboring RSU checks whether the information of the transaction is true or not before starting the consensus process. When the RSU detects the malicious behavior of the vehicle platoon, the RSU broadcasts that the message of the transaction is invalid to other RSUs. As the coins spent by the vehicle platoon are not included in the local one-day blockchain, their reputation score is lowered according to the beta reputation function and it constrains their activities.

6) Resistance Against Modifying and Stealing Coins: The malicious vehicle can attempt to reuse the coins that have already been spent. Every coin has its unique identification number and the history of coin usage is recorded transparently in the local one-day message blockchain. According to the record in the blockchain, if the vehicle uses the same voting coin twice, only one vote will be allowed and accepted. It is also impossible for the vehicle to forge voting coins because all voting coins, spent by the vehicle in the local one-day message blockchain, are originally sent by the RSU when the vehicle joins the local network. The RSU and the other vehicles can verify whether the RSU has sent the voting coin to the vehicle or not, and if there is no transaction record, it is discovered that the voting coin is a counterfeit coin. A malicious vehicle also can attempt to steal the coin in order to cheat and use the other vehicle's coin as its own coin. However, the linking of coin usage is stored in the local one-day message blockchain, as mentioned above. Therefore, the owner of the coin is recorded in the local one-day blockchain, and nobody can use the coins owned by other vehicles.

7) Limiting the Activity of Malicious Vehicles: Once a vehicle has done something malicious, there is a high probability that it will repeat the misbehavior again later. To prevent the malicious vehicle's impact on the network, the RSU determines the reputation score according to the behavior of the vehicle and hands out the voting coins based on the updated reputation score.

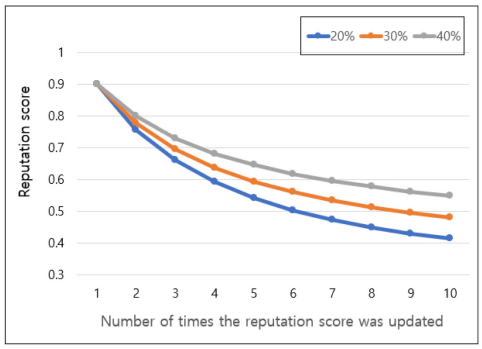


Fig. 3. Reputation score when coins, used by a high-reputed vehicle, are included in the block at a low rate.

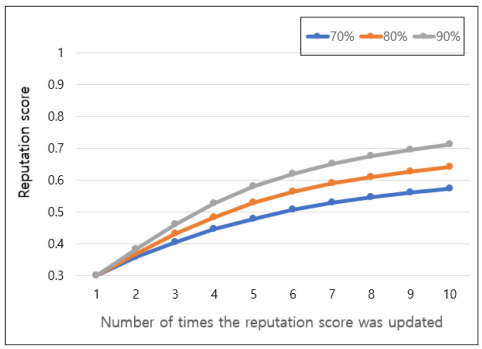


Fig. 4. Reputation score when coins, used by a low-reputed vehicle, are included in the block at a high rate.

Assuming that the vehicle always uses 80% of its own coin, when the vehicle's reputation score is 0.9 and the accumulated amount of spent coins is 300, the reputation score will decrease if less than half of vehicle's spent coins are included in the block. As shown in Fig. 3, when the used coins of 20%, 30%, and 40%, are included in the block and the reputation scores are recalculated nine times, the reputation scores are lowered to 0.56, 0.49, and 0.42, respectively. As a result, although the reputation score is high, if the activity is unconscionable, the reputation score decreases. Vehicles that do not behave in good faith in the network will have relatively few voting coins, allowing them to vote with fewer $ReportingM$ and propagate fewer message transactions than other honest vehicles.

Conversely, if a low-reputed vehicle continuously increases the amount of coins included in the block, the beta reputation function can increase its reputation since the beta reputation score is determined according to the ratio of the accumulated amount of coins included in the block and the accumulated amount of coins not included in the block of the vehicle. When 70%, 80%, and 90% of the coins used by vehicles with low reputation scores of 0.3 are continuously stored in the blockchain, the reputation scores all rise above 0.5 after six reputation recalculations as shown in Fig. 4. Therefore, even if the vehicle's previous reputation score is low, our proposed system can motivate the low-reputed vehicle to actively cooperate in reporting and voting to increase their reputation score.

TABLE III
PARAMETERS

Parameters	Settings
Playground	3000*3000
lane	2
Minimum signal reception threshold	-110 dBm
Maximum transmission range	800 m
Transaction verification time	0.83 ms
Distance between RSU	1440 m
Transaction Signature Time	140 ms
Simulation time	500 s
Mac protocol	IEEE 802.11p

V. Simulation and Performance Analysis

In this section, we evaluated our local one-day blockchain network through simulations to validate its effectiveness. We used python IDE [31] and Omnet++ [32] to simulate the vehicular networks. We implemented our proposed protocol using the Bitcoin-Python library [33] and Veins simulation [34], which consists of SUMO [35] and Omnet++. We assumed that the computing power of the vehicles matches a Raspberry Pi 3, and the computing power of an RSU is the same as a laptop. The parameters and settings information for our simulation are shown in Table III.

A. Performance Analysis

1) Performance Analysis of LPBFT: In order to demonstrate the efficiency of our proposed LPBFT, we compared Practical Byzantine Fault Tolerance (PBFT) and LPBFT in the same simulation environment. According to the existing assumptions of the PBFT algorithm, when f is the maximum number of malicious RSUs, the total number of RSUs, which are participating for consensus, should be $3f + 1$. While the primary node of the PBFT is randomly selected, the primary node of the LPBFT is flexibly selected according to the accident location stored in the message transaction. In the simulation environment, we assumed that the RSUs were placed one by one at regular intervals on a long road, such as a highway, and each RSU could communicate with the RSUs located next to it. We set up the total number of RSUs starting from 4, when f is 1, and increasing up to 100. The transaction verification time is a value obtained by assuming that 5 $VotingM$ are included in a message transaction. The RSU's transmission coverage is 800 m and the distance between RSUs was set to 1440 m when the RSU is located so that the transmission range overlaps by 0.8. according to the [36]. Each transaction includes 5 $VotingM$ in this simulation.

We first simulated the message propagation time depending on how many RSUs the message transaction had to pass. We assumed that it took 5 ms plus some random delay when the vehicle sent a message to the RSU.

When using a PBFT, the vehicle creates a message transaction and sends it to the nearest RSU, which then propagates the transaction until it is delivered to the primary RSU. So, if the primary RSU is far away from the nearest RSU to the vehicle, the delivery time increases. In the case of LPBFT, since the

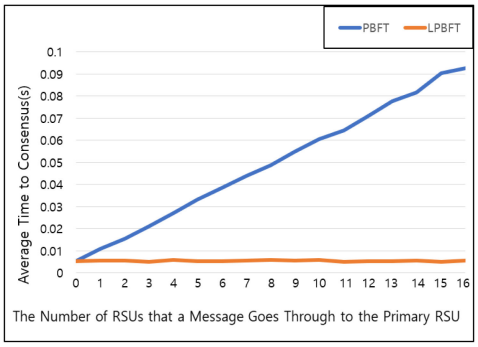


Fig. 5. Comparison of LPBFT and PBFT for Transmission Time before consensus algorithm starts.

RSU nearest to where the vehicle generates the message is determined to be the primary node, the time for delivering the message to the primary RSU is always constant. As shown in Fig. 4, the time that it takes for the message transaction of a vehicle to propagate to the primary RSU is always constant when applying LPBFT, while the time continuously increases when more RSUs are located between the primary node and the vehicle when the PBFT is used. It means that the larger the distance between the primary node and the vehicle, the longer the message transmission process is in PBFT. Fig. 5 shows the result of comparing the average consensus algorithm times of PBFT and LPBFT in vehicular networks. We conducted the simulation of PBFT by designating the fourth RSU as the primary node and specifying different RSUs to receive the vehicle's message transaction each time. But, if we choose the 5th or 6th or farther away node as the primary node, the message transmission time will be increased. As shown in Fig. 4, LPBFT is more efficient in consensus compared to PBFT in a vehicular network situation. When the number of RSUs is 4, since the average time of consensus is 0.0506 s with PBFT and 0.046 s with LPBFT, the time difference is within 0.012 s. However, when the number of RSUs increased to 100, it took about 1.26 s for PBFT and 0.72 s for LPBFT, which shows that PBFT algorithm is 1.42 times slower than LPBFT algorithm. As we mentioned above, in the case of PBFT, when the vehicle transmits a transaction to the neighboring RSU, the RSU spends time communicating until the designated primary RSU receives the transaction. However, LPBFT achieves a more efficient consensus process because the RSU closest to the incident location becomes the primary RSU for the incident transaction. For this reason, the LPBFT algorithm appears to be a suitable consensus algorithm in vehicular networks.

2) Complexity Analysis of Proposed Protocol: Among blockchain-based vehicle reputation models, the previous studies that considered vehicular privacy are Bars [13] and Creditcoin [19]. The Bars system uses the CA to reissue a public key of the vehicle each time it enters the vehicular network, which is the most simplistic way to handle the privacy of the vehicle. So, the overhead of the CA to reissue and manage the temporary public keys is quite high and quite time consuming. Moreover, it is not a decentralized approach, but one that depends entirely on the CA for barely adequate privacy. However, both the Creditcoin [19] and our scheme attempt to solve the privacy issue using a decentralized approach including RSU and neighboring vehicles. So, we compared the time complexity of the two decentralized approaches, our system and Creditcoin [19], while excluding Bars [13], in order to evaluate their performance.

TABLE IV
COMPUTING OVERHEAD

Model	Entity	Time Complexity		
		Request	Reply	Verification
Creditcoin	Vehicle	*O(cn)*	*O(1)*	*O(n)*
	RSU	-	-	-
Proposed Model	Vehicle	*O(1)*	*O(1)*	-
	RSU	-	-	*O(n)*

The results of the time complexity analysis are shown in Table IV . We counted the number of elliptic curve point addition operations required in each phase. It is relatively time consuming compared to other elliptic curve arithmetic operations. Creditcoin [19] needs to generate the public key of the forgery Identities to be used for the threshold ring signature and uses the combined-public keys, which require the vehicle or RSU to perform the elliptic curve point addition operations that are the same size as the public key vector. The time complexity of the request phase in Creditcoin [19] is $O(cn)$, where c is proportional to the length of public key. Our proposed model uses ECDSA, which has 3 point elliptic curve point additions for signing, so the time complexity is $O(1)$. This means that the increase doesn't depend on the size of the number of vehicles; it is always the same which is an advantage of our system. The time complexity of both models in the reply phase is equal to $O(1)$. In the verification phase, since the signatures of the vehicles that reply to a request should be verified, the time complexity of the verification phase is $O(n)$ in both Creditcoin [19] and our system. However, in our proposed model, the RSU has the role of transaction verification, while the vehicles, which have relatively low computing power compared to the RSU, verify the signatures of the transactions in Creditcoin [19]. This can also be an overhead problem for the vehicle in the Creditcoin [19]. This means that our proposed model is more efficient and practical than Creditcoin [19] in protecting the vehicular privacy.

B. Simulation Results

In order to evaluate the performance of the proposed local one-day message blockchain network, we designed a virtual traffic environment using Veins simulation and performed simulations for highway and surface street grid scenarios. Transaction signature time measures the time it takes for the vehicle to sign the reply message in the Reply step. As we assumed that the computational power of the vehicle is the same as that of the Raspberry Pi 3, it was set to 140 ms, which is the time it took to create an ECDSA signature by using bitcoin 1.1.42 library [33] in the Raspberry Pi 3. We designed the highway and surface street scenario environments by referencing the map of the Korean metropolitan area. In Fig. 7(a) is a map of Seoul Oegwaksunhwan highway, (b) is a highway map drawn

TABLE V
THE REQUIRED NUMBER OF REPLY MESSAGES FOR EACH SCENARIO

Num	Road Type	Traffic Condition	Vehicle's Average Velocity (km/h)	Safety Distance (m)	The Number of Neighboring Vehicles	Required Number of Reply Messages	A Reporting Message Verification Time (seconds)	
							PC (intel core i5)	Raspberry Pi 3
1	Highway	Heavy	40	50	47	36	0.01	0.13
2		Light	90	100	26	20		
3	Surface Road	Heavy	25	35	63	48		
4		Light	60	70	35	27		

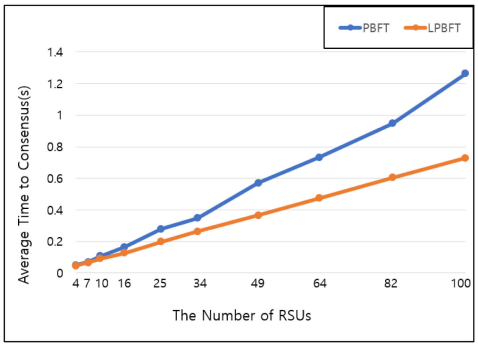


Fig. 6. Comparison of PBFT and LPBFT for Consensus.

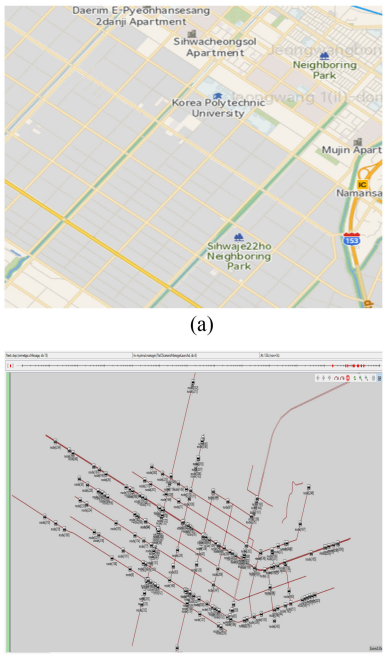
(a)

(b)

Fig. 7. Simulation Scenario of Highway. (a) A map of Jeongwang-dong surface street. (b) A map of a grid-shaped surface street in Omnet++.

in Omnet++ based on this. In Fig. 8(a) is the grid-shaped Jeongwang-dong surface street and (b) is a grid map based on this. In the simulation, the vehicle's route was random, which is generated using SUMO, and the minimum distance between the vehicles was specified according to the vehicle safety distance standard as designated by the Road Traffic Authority [37]. We simulated four traffic scenarios: 1) heavy traffic on the highway, 2) light traffic on the highway, 3) heavy traffic on the surface street grid, and 4) light traffic on the surface street grid. We set the vehicle length to 5 m and assumed that this is a safe distance between vehicles. By using these parameters, we could calculate the maximum number of vehicles that can exist in each simulated scenario. Since the safety distance changes according to the speed of the vehicle, the number of vehicles was set differently for each scenario. We simulated the movement of vehicles for a simulation period of 500 s in each scenario and a random vehicle broadcasts a new request message every 1 s simulation period. To prevent a malicious vehicle platoon from generating a valid amount of replies to a false request, it is necessary to have more than a certain percentage (ex. $\frac{3}{4}$ according to our consensus algorithm, LPBFT) of the total number of vehicles that can send reply vote. We define a credibility factor (cf) as the ratio of the reply messages received by the requesting vehicle from neighboring vehicles to each other within an 800 m radius of the requesting vehicle. This indicates the reliability of the request message, and the cf of all request messages should be set at 0.5 or above for secure message transaction generation. We assume that if the cf value is more than 0.75, the message transaction is sufficiently reliable and directly incorporated into the transaction pool of the local one-day message blockchain.

Table V shows the average speed and safety distance of the vehicle for each scenario, the number of vehicles that can exist within an 800 m radius of the vehicle, and the number of replies required. The verification time in the table is the time required for a request vehicle to verify the reply messages. On a highway, when there is a traffic jam, the average vehicle velocity is under 40 km/h so the recommended safety distance between the vehicles is up to 50 m, and accordingly, there can be at least about 47 vehicles within an 800 m radius of the vehicle. If the highway traffic is light, the vehicle velocity will be 90 km/h and the recommended safety distance between vehicles is 100 m. The expected number of vehicles that can exist around the vehicle is then 26. In order for 75 percent of vehicles to agree on a request message (i.e. $cf > 0.75$), the requesting vehicle needs to collect more than 36 replies in heavy highway traffic and 20 in light highway traffic. On a surface road, vehicles are spaced up to 35 m apart in heavy traffic and 70 m apart in light traffic, according to traffic safety guidelines. In order to calculate the number of neighboring vehicles on surface roads, the intersection needs to be excluded as vehicle movement there is irregular. Therefore, there can be a possible 63 neighboring vehicles on a busy surface road, while an expected 35 vehicles exist on surface roads with light traffic. It is recommended for request vehicles to collect 48 replies in heavy traffic on surface roads and 27 replies in



(a)

(b)

Fig. 8. Simulation Scenario of Surface Street.

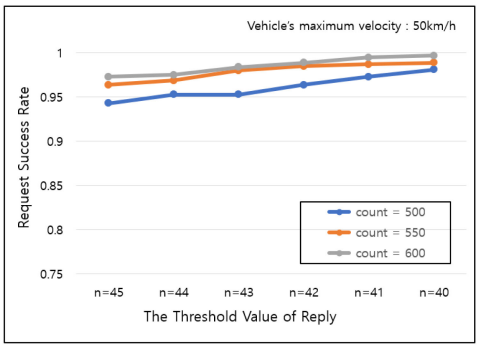


Fig. 9. Request Success Rate in highway when the traffic is heavy.

light traffic on surface roads in order to guarantee that the cf is over 0.75. The time needed to verify the required number of replies in each traffic scenario is less than 3 s, which is acceptable for protocol operation. The results of the simulation for each scenario are as follows. The request success rate is the rate at which vehicles have succeeded in collecting $VotingM$ above the threshold value.

1) Heavy Traffic on the Highway: According to the ITS National Transport Information Center [38], the congestion standard on highways is when the vehicle speed is less than 40 km/h. Therefore, we assumed that traffic on the highway was congested when the vehicle's maximum speed to 50 km/h. In this scenario, we simulated 500, 550, and 600 counts, respectively. The count is the number of vehicles that can exist in a 1 km road per hour.

Fig. 9 shows the rate at which the vehicle successfully performed a request according to N, which is the threshold of reply messages from neighboring vehicles needed by the vehicle for a successful request. If the N threshold is set higher, the reliability of the request message may increase, but the probability of

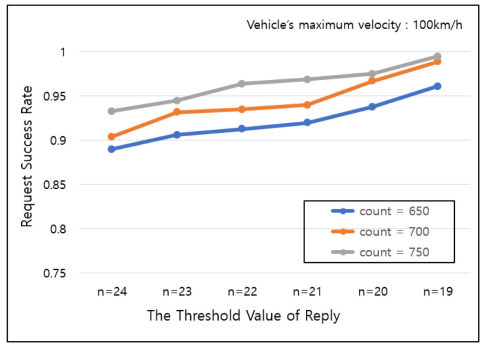


Fig. 10. Request Success Rate in highway when the traffic is light.

collecting all N replies can decrease. When the counts are 600 and 550, the request success rate remains high above 0.95 until N becomes 45. The success rate of 500 counts is slowly falling under 0.95 where N is 45, but the request success rate remains above 0.9. Therefore, if N is set to 45, a high success rate and cf can be satisfied in heavy traffic on the highway. Due to heavy traffic, the speed of the vehicles on the highway is low and the distance between the vehicles is small, so it is possible to get enough reply messages from neighboring vehicles even when the N threshold is set at a high number.

2) Light Traffic on the Highway: ITS National Transport Information Center [38] said that the standard for smooth traffic is when the vehicle speed is more than 80 km/h on highways. We assumed that the vehicle's maximum speed is 100 km/h in light traffic. We simulated 650, 700, and 750 counts, respectively. In Fig. 10, when the count is 650 and N is 23, the request success rate is barely over 0.9. Therefore, in order for the request success rate to be 0.9 or higher for securely operating blockchain model, N should be at least 23. According to Table V, for cf to reach 0.75, the threshold value of N is 20. This parameter is applicable to the protocol as shown in Fig. 10. In the second scenario, it is more difficult to collect a large number of reply messages because the number of vehicles on the road is smaller and the speed is faster than in the first scenario. Therefore, when compared with Fig. 9, N should be smaller when traffic is light to keep the request success rate high.

3) Heavy Traffic on the Surface Street: The congestion standard on a regular road is when the vehicle speed is less than 30 km/h in [38]. We set the vehicle's maximum speed to 35 km/h. In this scenario, We simulated 400, 450, 500 count, respectively. As you can see in Fig. 11, when The count is 500 or 450 on a surface street, the request success rate remains high to about 0.9 when N is 48. When the threshold value is 48 or less, it will satisfy a high success rate and cf in heavy traffic on the surface street. Compared to the simulation results of highways with heavy traffic, it can be seen that despite the lower number of vehicles participating in the simulation, the request success rate on the surface street grid is almost similar. For lattice surface roads, multiple different roads exist in the radius around the vehicle, making them easier to collect replies than straight highway.

4) Light Traffic on the Surface Street: If the vehicle moves faster than 50 km/h, traffic is running smoothly according to [38].

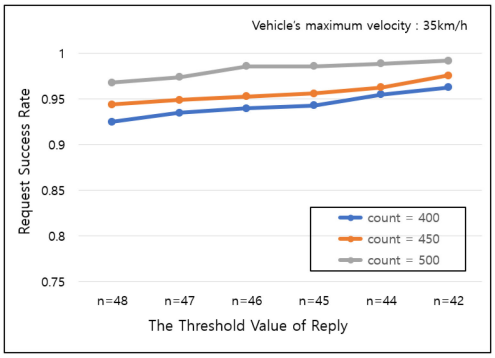


Fig. 11. Request Success Rate in surface street when the traffic is heavy.

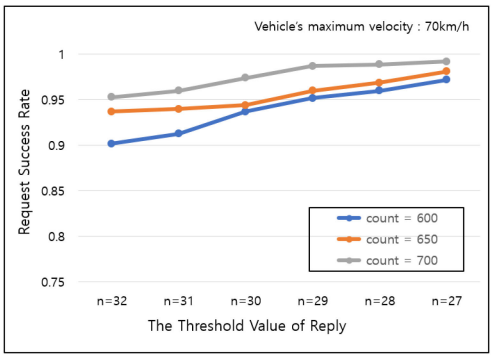


Fig. 12. Request Success Rate in surface street when the traffic is light.

Therefore, we assumed that traffic on the surface street grid was light when the vehicle's maximum speed was 70 km/h. We simulated 600, 650, 700 counts, respectively. Fig. 12 shows the request success rate when the traffic is light on the surface street. When we set the threshold N at 32 or less, the request success rate in all three cases (600, 650, 700 counts) is over 0.9. Compared to the simulation results on light traffic in highways, Fig. 12 shows that more replies are collected from simulations on surface streets despite similar traffic. This difference is because the speed of highway light traffic can be much faster and more consistent than traffic on a surface street with light traffic due to the lack of interference.

The simulation results for each scenario show that it is feasible to receive a sufficient number of replies for the high reliability of the message. The threshold value N should be applied differently according to different road types, vehicle speeds, and numbers, so RSU can flexibly apply the threshold value N according to road conditions.

VI. Conclusion

In this paper, we proposed a two layered blockchain-based reputation system in vehicular networks. By applying a local one-day message blockchain, we fulfilled the goals of managing real time traffic messages effectively and protecting partial privacy of vehicles. To support fast block propagation in a local one-day message blockchain, we suggested a new consensus algorithm, named LPBFT and applied it to a local one-day message blockchain. We simulated highway and surface street scenarios using the local one-day message blockchain system. Its result demonstrated the practicality of the system's application in real world traffic situations. Through our proposed system, vehicles share reliable real-time traffic messages with each other, based on a reputation system that does not reveal their true identities but still guarantees vehicle trustworthiness. In addition, as vehicles continue to participate in the system, their reputation scores change based on their activity in a local one-day blockchain. The global vehicle reputation blockchain stores the reputation score, which can be used as an incentive for honest vehicle activities, and which can then be helpful for operating a safe transportation system. In our two layered blockchain-based reputation system, the reputation score of each vehicle is calculated based on their coin usage. In order to strengthen the trust level of the reputation system, we will study the vehicle reputation model that considers other external factors such as opinions of surrounding vehicles and vehicle activity patterns for more accurate reputation calculations in future studies.

References

[1] G. Dimitrakopoulos and P. Demestichas, "Intelligent transportation systems," *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 77–84, Mar. 2010.
[2] Z. Yang, K. Zheng, K. Yang, and V. C. Leung, "A Blockchain-based reputation system for data credibility assessment in vehicular networks," in *Proc. IEEE 28th Ann. Int. Symp. Pers., Indoor, Mobile Radio Commun.*, 2017, pp. 1–5.
[3] R. G. Engoulou, M. Bellaiche, T. Halabi, and S. Pierre, "A decentralized reputation management system for securing the internet of vehicles," in *Proc. Int. Conf. Comput., Netw. Commun.*, 2019, pp. 900–904.
[4] W. Li and H. Song, "ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 960–969, Apr. 2016.
[5] C. Chen, J. Zhang, R. Cohen, and P.-H. Ho, "A trust modeling framework for message propagation and evaluation in VANETs," in *Proc. 2nd Int. Conf. Inf. Technol. Convergence Serv.*, 2010, pp. 1–8.
[6] O. Chikhaoui, A. B. C. Douss, R. Abassi, and S. G. El Fatmi, "Towards a privacy preserving and flexible scheme for assessing the credibility and the accuracy of safety messages exchanged in VANETs," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, 2018, pp. 1–9.
[7] G. Costantino, F. Martinelli, I. Matteucci, A. Bertolino, A. Calabro, and E. Marchetti, "Cars: Context aware reputation systems to evaluate vehicles' behaviour," in *Proc. 26th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process.*, 2018, pp. 446–453.
[8] J. Oluoch, "A distributed reputation scheme for situation awareness in vehicular Ad Hoc networks (VANETs)," in *Proc. IEEE Int. Multi-Disciplinary Conf. Cogn. Methods Situation Awareness Decis. Support*, 2016, pp. 63–67.
[9] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25408–25420, 2017.
[10] M. A. Azad, S. Bag, S. Parkinson, and F. Hao, "TrustVote: Privacy-preserving node ranking in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 5878–5891, Aug. 2019.
[11] S. Tangade, S. S. Manvi, and P. Lorenz, "Trust management scheme based on hybrid cryptography for secure communications in vanets," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5232–5243, May 2020.
[12] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Bus. Inf. Syst. Eng.*, vol. 59, no. 3, pp. 183–187, 2017.
[13] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A privacy-preserving trust model based on Blockchain for VANETs," *IEEE Access*, vol. 6, pp. 45655–45664, 2018.
[14] R. Shrestha, R. Bajracharya, A. P. Shrestha, and S. Y. Nam, "A new type of Blockchain for secure message exchange in VANET," *Digital Commun. Netw.*, vol. 6, no. 2, pp. 177–186, May 2020.
[15] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[16] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.
[17] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.
[18] F. Kandah, B. Huber, A. Skjellum, and A. Altarawneh, "A Blockchain-based trust management approach for connected autonomous vehicles in smart cities," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf.*, 2019, pp. 0544–0549.
[19] L. Li *et al.*, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018.
[20] S. Gao, G. Piao, J. Zhu, X. Ma, and J. Ma, "Trustaccess: A trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5784–5798, Jun. 2020.
[21] B. Chen, L. Wu, H. Wang, L. Zhou, and D. He, "A blockchain-based searchable public-key encryption with forward and backward privacy for cloud-assisted vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5813–5825, Jun. 2020.
[22] S. Iqbal, A. W. Malik, A. U. Rahman, and R. M. Noor, "Blockchain-based reputation management for task offloading in micro-level vehicular fog network," *IEEE Access*, vol. 8, pp. 52968–52980, 2020.
[23] M. Castro, B. Liskov, "Practical byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Des. Implementation*, vol. 99, no. 1999, 1999, pp. 173–186.
[24] N. Magaia and Z. Sheng, "Refiov: A novel reputation framework for information-centric vehicular applications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1810–1823, Feb. 2019.
[25] O. Chikhaoui, A. B. Chehida, R. Abassi, and S. G. El Fatmi, "A ticket-based authentication scheme for VANETs preserving privacy," in *Proc. Int. Conf. Ad-Hoc Netw. Wireless*, 2017, pp. 77–91.
[26] R. Shrestha, S. Y. Nam, R. Bajracharya, and S. Kim, "Evolution of V2X communication and integration of Blockchain for security enhancements," *Electronics*, vol. 9, no. 9, 2020, Art. no. 1338.
[27] N. Gallego, A. Mocholi, M. Menendez, and R. Barrales, "Traffic monitoring: Improving road safety using a laser scanner sensor," in *Proc. Electron., Robot. Automot. Mechanics Conf.*, 2009, pp. 281–286.
[28] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," *ICT Exp.*, vol. 6, no. 2, pp. 93–97, Jun. 2019.
[29] R. Ismail and A. Josang, "The beta reputation system," in *Proc. 15th Bled Electron. Commerce Conf.*, 2002, pp. 2502–2511.
[30] J. H. Silverman and J. Suzuki, "Elliptic curve discrete logarithms and the index calculus," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 1998, pp. 110–125.
[31] *Idle-python 3.8.3 documentation*, Accessed: May 30, 2020. [Online]. Available: https://docs.python.org/3/library/idle.html
[32] A. Varga, "Omnet" in *Modeling and Tools for Network Simulation.* Berlin, Germany: Springer, 2010.
[33] "Bitcoin. pypi," Accessed: Apr. 2, 2020. [Online]. Available: https://pypi.org/project/bitcoin
[34] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.
[35] P. A. Lopez *et al.*, "Microscopic traffic simulation using sumo," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2575–2582.
[36] J. Lee and C. M. Kim, "A roadside unit placement scheme for vehicular telematics networks," in *Advances in Computer Science and Information Technology*. Berlin, Germany: Springer, 2010.
[37] "Road traffic authority," Accessed:Apr. 2020, [Online]. Available: https://www.koroad.or.kr/kp_web/index.do
[38] "Its national transport information center," Accessed : Apr. 23, 2020. [Online]. Available: http://www.its.go.kr/

Soojin Lee (Student Member, IEEE) received the B.S. degree in 2019 from the Division of Electrical Engineering, Hanyang University ERICA Campus, Ansan, South Korea, where she is currently working toward the master's degree with the Department of Electrical Engineering. Her research interests include blockchain, IoT security, and privacy protection.

Seung-Hyun Seo (Member, IEEE) received the B.S. degree from the Department of Mathematics, Ewha Womans University, Seoul, South Korea, in 2000, and the M.S. and Ph.D. degrees in computer science from Ewha Womans University, Seoul, South Korea, in 2002 and 2006, respectively. She is currently a Professor with Hanyamg University. Before joining the faculty with Hanyang University, in 2017, she was an Assistant Professor with Korea University Sejong campus, South Korea, for two years. Before that, she was a Postdoctoral Researcher of computer science with Purdue University, West Lafayette, IN, USA, for two and half years, a Senior Researcher of Korea Internet and Security Agency, Seoul, South Korea, for two years, and a Researcher for three years with Financial Security Agency, South Korea. Her main research interests include cryptography, the IoT security, mobile security, blockchain, and post-quantum cryptography. Her research interests include blockchain, IoT security, and privacy protection.