TRUCON: Blockchain-Based Trusted Data Sharing With Congestion Control in Internet of Vehicles

Mingyang Yuan, *Graduate Student Member, IEEE*, Yang Xu, *Member, IEEE*, Cheng Zhang, *Student Member, IEEE*, Yunlin Tan, Yichuan Wang, *Member, IEEE*, Ju Ren, *Senior Member, IEEE*, and Yaoxue Zhang, *Senior Member, IEEE*

***Abstract*— The Internet of vehicles (IoV) has a substantial impact on traffic efficiency improvement and accidents avoidance. Due to restricted resources, vehicles must share observed data with RSUs and other vehicles to execute some time-tolerant computing tasks. However, data provided by vehicles cannot always be trusted due to the presence of attackers. Fake messages could have catastrophic ramifications, such as vehicle collisions. Furthermore, extensive data sharing might cause channel congestion, resulting in the loss of vital messages during delivery. To overcome the aforementioned issues, we propose TRUCON, a blockchain-based trusted data sharing mechanism with congestion control in IoV. Firstly, we propose a Kademlia algorithm-based traffic data forwarding method to control channel congestion state. By adjusting the bucket size and distance threshold, source vehicles can limit the number of reference vehicles forwarded. Secondly, we present a cuckoo filter-based traffic data deduplication and discrimination approach. To avoid repetitive sharing, vehicles and RSUs can check their local filters to verify if the current data report has been shared. Based on the foregoing, we propose a blockchain-based trust management mechanism with congestion control. RSUs serve as full nodes while vehicles are light nodes in the blockchain. Finally, we develop a trust management prototype system with congestion control that incorporates both on-chain and off-chain parts. It signifies that our scheme is both feasible and effective.**

Manuscript received 9 June 2022; revised 17 October 2022; accepted 28 November 2022. Date of publication 15 December 2022; date of current version 1 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62272154 and Grant 62002113, in part by the Natural Science Foundation of Hunan Province under Grant 2021JJ40122, and in part by the Open Project Program of Key Laboratory of Blockchain and Cyberspace Government of Zhejiang Province under Grant 202110138. The Associate Editor for this article was S. Wan. *(Corresponding author: Yang Xu.)*

Mingyang Yuan is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: ymy1999@csu.edu.cn).

Yang Xu and Cheng Zhang are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, Zhejiang University, Hangzhou 310027, China (e-mail: xuyangcs@hnu.edu.cn; zhangchengcs@hnu.edu.cn).

Yunlin Tan is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: tanyunlin@hnu.edu.cn).

Yichuan Wang is with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China, and also with the Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China (e-mail: chuan@xaut.edu.cn).

Ju Ren and Yaoxue Zhang are with the Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing 100084, China (e-mail: renju@tsinghua.edu.cn; zhangyx@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3226500

***Index Terms*— Internet of Vehicles, blockchain, trust management, data sharing, congestion control.**

I. Introduction

ACCORDING to WHO estimates, road traffic accidents kill 1.35 million people worldwide every year and are currently the leading cause of mortality among children [1]. The Internet of vehicles (IoV) provides a solution to this. It allows vehicles to communicate and share data with each other, which enables the cooperative intelligent transportation systems (C-ITS) [2]. So every vehicle on the road has a broader view of traffic information. As a result, IoV can avoid possible accidents and arrange vehicle movement ahead of time to enhance traffic conditions through data collection and forecasting [3]. Due to restricted resources in computing and storage, vehicles will outsource some delay-tolerant tasks to cloud service provider (CSP).

However, data shared by vehicles cannot be trusted always due to the characteristics of the open environment and rapid mobility of IoV [4]. Vehicles may be attacked and then broadcast fake messages to others, which has a significant impact on how other vehicles make decisions. For example, they claim that there is a green light ahead but it's actually a red one. These fake messages can lead to serious consequences such as vehicle collisions sometimes [5]. Therefore, it is vital for every vehicle to have the ability to discriminate the correctness of information from others. To ensure the credibility of traffic data, some scholars propose to establish trust management system in IoV [6]. At first, trust values of vehicles are usually generated according to their behaviors and stored in trusted authority. But these solutions rely on centralized nodes which results in problems such as substantial maintenance cost and single point of failure.

The emergence of blockchain technology [7] provides a solution to the above problems. It stores data in blocks which are linked together one by one and implements complex functions using smart contracts. Reference [8], [9] Due to the characteristics of decentralization, tamper resistance and traceability, blockchain is now widely used in various fields [10], [11]. Thus, some researchers want to introduce blockchain into trust management in IoV. Yang et al. [12] propose that vehicles judge the correctness of data provided by

others through the bayesian inference model and give ratings to them. These ratings will be calculated as trust value offset and recorded on blockchain by RSUs. However, messages are usually broadcast to other vehicles in these schemes. It causes communication channel congestion and blocks the transmission of important messages sometimes.

To solve the above problems, our scheme reduces unnecessary communication overhead in two major ways. For one thing, the further two vehicles are, the less likely they are to observe the same traffic information. To reduce ineffective trust evaluation times, we propose a Kademlia algorithm [13] based traffic data forwarding mechanism in IoV. We assign every active vehicle with a coordinate combining with area code and vehicle code according to their real geographic location. By measuring the distance between vehicles with their coordinates, every vehicle can construct a routing table. The amount of data forwarded can be adjusted according to bucket size k and our preset distance threshold d. It can effectively reduce the channel communication overhead.

For another, chances are that vehicles upload duplicate data to RSUs due to the same traffic conditions observed. To alleviate the communication overhead of this part, we propose a traffic data deduplication and discrimination method based on the cuckoo filter. RSUs will verify the local filter to see if the report has previously been uploaded, and then it will record the number of repetitions on the filter. When conflicting reports appear, the credibility of each report can be judged according to their repeat times. For the vehicle side, they cache the filter from corresponding RSUs for a period of time. They use the local filter to reduce redundant reports uploading the same as RSUs do.

The contributions of this paper are summarized as follows.

- We propose a Kademlia algorithm-based traffic data forwarding method in IoV. By adjusting bucket size k and distance threshold d, we can limit the number of reference vehicles being forwarded to control the channel congestion state.
- We present a cuckoo filter-based traffic data deduplication and discrimination approach. Both RSUs and vehicles check their local filters to see if the traffic reports have been uploaded to avoid repetitive sharing. Every report's repeat time is also recorded to assess its credibility.
- We propose TRUCON, a blockchain-based trust management mechanism with congestion control in IoV. RSUs serve as full nodes while vehicles are light nodes. Our scheme reduces communication overhead while assuring data accuracy when compared to existing solutions.
- On the basis of the foregoing, we develop a trust management prototype system with congestion control that incorporates on-chain and off-chain parts. We also conduct relevant experiments under various settings to ensure that our method is effective and feasible.

The remainder of this paper is summarized as follows. Section II states the related work. Section III introduces the technologies used in our scheme. Section IV defines the relevant system model definitions. In Section V, we propose TRUCON, a blockchain-based trusted data sharing mechanism with congestion control in IoV. Then, we conduct a detailed security analysis in Section VI. Section VII describes relevant designs and results of our experiments. The last Section VIII concludes this paper and discusses future work.

II. Related Work

In this section, we make a brief summary and comparison of existing trust management and communication congestion control mechanisms in IoV.

A. Trust Management Mechanism in IoV

In order to ensure the safety of vehicles, scholars propose to establish a trust management mechanism to guarantee the quality of traffic data. Centralized schemes are first proposed in a large number of studies [14], [15]. SIRC [16] is a protocol establishing a reputation system to encourage cooperative downloading in highway VANETs. By helping with the client vehicle in file downloading successfully, the proxy vehicle will obtain credit, which motivates the system to keep running. Garcia et al. [17] proposed a trust management mechanism to identify and isolate the hijacked vehicles. The reputation information is generated by another vehicle and stored in certification entity. To prevent tampering, all the reputation messages are attached with digital certificates. These centralized schemes work efficiently but have issues with single point of failure and being tempered with.

To address this problem, decentralized trust management schemes are proposed [18], [19]. Yang et al. [12] introduced blockchain into trust management in vehicular networks first. In their scheme, vehicles can validate messages using the bayesian inference model. RSUs are responsible for collecting trust data from vehicles and maintaining a blockchain. To increase the transaction throughput, Singh et al. [20] presented a sharding blockchain based trust management scheme. It is adaptive for compatibility with existing misbehavior detection strategies. Revocation of misbehaving vehicles is also considered to achieve stability of the established system. Liu et al. [21] use blockchain and identity-based group signatures to achieve trust management along with conditional privacy preservation. It can also find out the identity of sender when receiving fake messages.

However, the above decentralized solutions cause massive communication overhead during trust data collecting process and might not be suitable for scenarios with heavy communication pressure. In our work, Kademlia algorithm is introduced to restrict the number of forwarded vehicles. So the communication overhead can be reduced.

B. Communication Congestion Control Mechanism in IoV

To solve the communication congestion problem brought by high-frequency data exchange in IoV, massive studies have been published [22]. The mainstream solutions can be divided into two groups. One of them focuses on applying methods such as reducing data transfer latency and increasing channel capacity to improve network channel quality. LACC [23] is a communication congestion control protocol that adopts

linear integer programming instead of a greedy approach to the neighbor selection process in IoV. However, LACC is designed for short-distance data transmission. To optimize the long-distance communication process, Zhioua et al. [24] put forward a multi-metric QoS balancing gateway selection algorithm to elect a better gateway to connect the source vehicle to the LTE advanced infrastructure (V2I). In this way, communication stability is improved a lot. Besides, a software-defined cooperative architecture in 5G and VANETs and a data-sharing algorithm based on graph theory are proposed by Luo et al. in [25] to improve data sharing performance.

The other kind of method is to reduce network traffic by slowing down the data generating rate at the source node. Zhuang et al. [26] propose a congestion-adaptive data collection scheme (CADC) that adjusts the data generating rate according to congestion level. Besides, they analyze the impact of congestion control on data accuracy. Wang et al. [27] propose a time division based dynamic clustering mechanism to achieve efficient data dissemination in bidirectional highways. Some researchers propose to introduce blockchain as well as access control methods to prevent unauthorized data sharing behavior to reduce data traffic [28], [29].

However, most solutions towards network channels improvement are limited in effect and difficult to realize. And it is a challenge to ensure accuracy and coverage of traffic data by slowing down data generating rate. In our scheme, cuckoo filter is used to avoid duplicate data uploading, which can effectively alleviate congestion while ensuring accuracy.

III. Preliminaries

A. Cuckoo Filter

The cuckoo filter is used to judge if a data report already exists under a large data volume. It is based on cuckoo hashing algorithm [30] that uses two hash functions and two hash tables to map data to the actual storage area. The basic unit of the cuckoo filter is called entry. To reduce the storage space for reports and provide the ability to store a large amounts of data, cuckoo filter stores fingerprints instead of complete data reports. Thus, cuckoo filter can support fast query and efficient deletion of large-scale data.

When a new report x comes, $h_1(x)$ and $h_2(x)$ are calculated respectively for indexing. Then one of the empty index positions is selected for insertion. If neither position is empty, randomly select an index and kick out the original report in that position and insert x. The kicked report repeated the above steps recursively, the algorithm process is shown in Fig. 1. To avoid falling into an infinite loop, cuckoo filter sets a threshold for the kick-out operation. If the threshold is exceeded, cuckoo filter needs to expand capacity or improve the hash algorithm to insert more reports.

B. Kademlia Algorithm

Kademlia algorithm [13] is an implementation of the distributed hash table (DHT), which aimed to quickly find the target resource without a central server in a P2P network. It stores information in the form of $< ID, Key >$, where ID is the unique identity of node and Key represents the identity

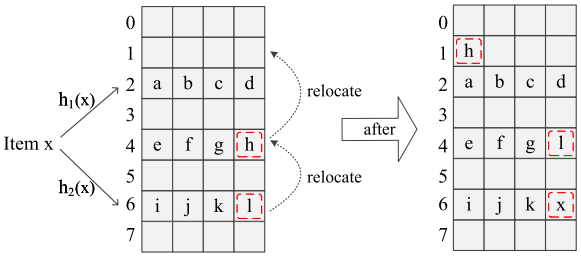


Fig. 1. Structure of cuckoo filter.

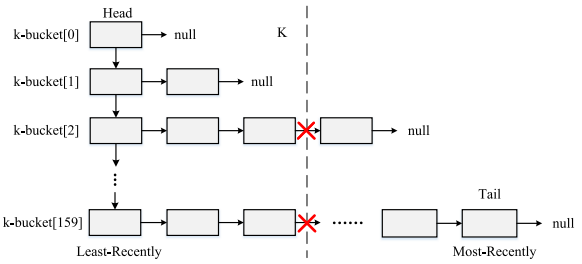


Fig. 2. k-bucket structure of Kademlia algorithm.

of resource. Kademlia algorithm organizes nodes as a binary tree and each leaf represents a node. Every node entering the network will be assigned an ID randomly.

Kademlia uses the XOR algorithm to calculate the distance between nodes and resources efficiently. As shown in Fig. 2, each node will construct a routing table structure to store the distance information, which is named k-bucket specifically. To look up the target node, the source node will calculate the XOR distance and find k nodes that are closest to the target node. By performing this process recursively, the target node will return the resource finally.

IV. System Model

A. Network Model

The architecture of TRUCON is shown in Fig. 3. Since the process of evaluating data quality has a great correlation with the distance between vehicles, we divide the whole road into several districts.

1) Vehicle: Vehicles are equipped with multiple sensors such as cameras, lidar, millimeter wave radar, and so on. These sensors will continuously generate data and merged it into a traffic data report according to the predefined format. To improve their trust value, vehicles will share the generated data reports. According to different responsibilities, vehicles can be source vehicles and reference vehicles. Every vehicle can be one or both of them.

- *Source vehicle:* Source vehicles generate traffic reports according to what they observed with equipped sensors. Then, they will share the reports with RSUs and reference vehicles.
- *Reference vehicle:* Reference vehicles receive traffic data reports from source vehicles and give ratings according to their local observed data. As a light node, they will invoke the smart contract and upload the trust values.

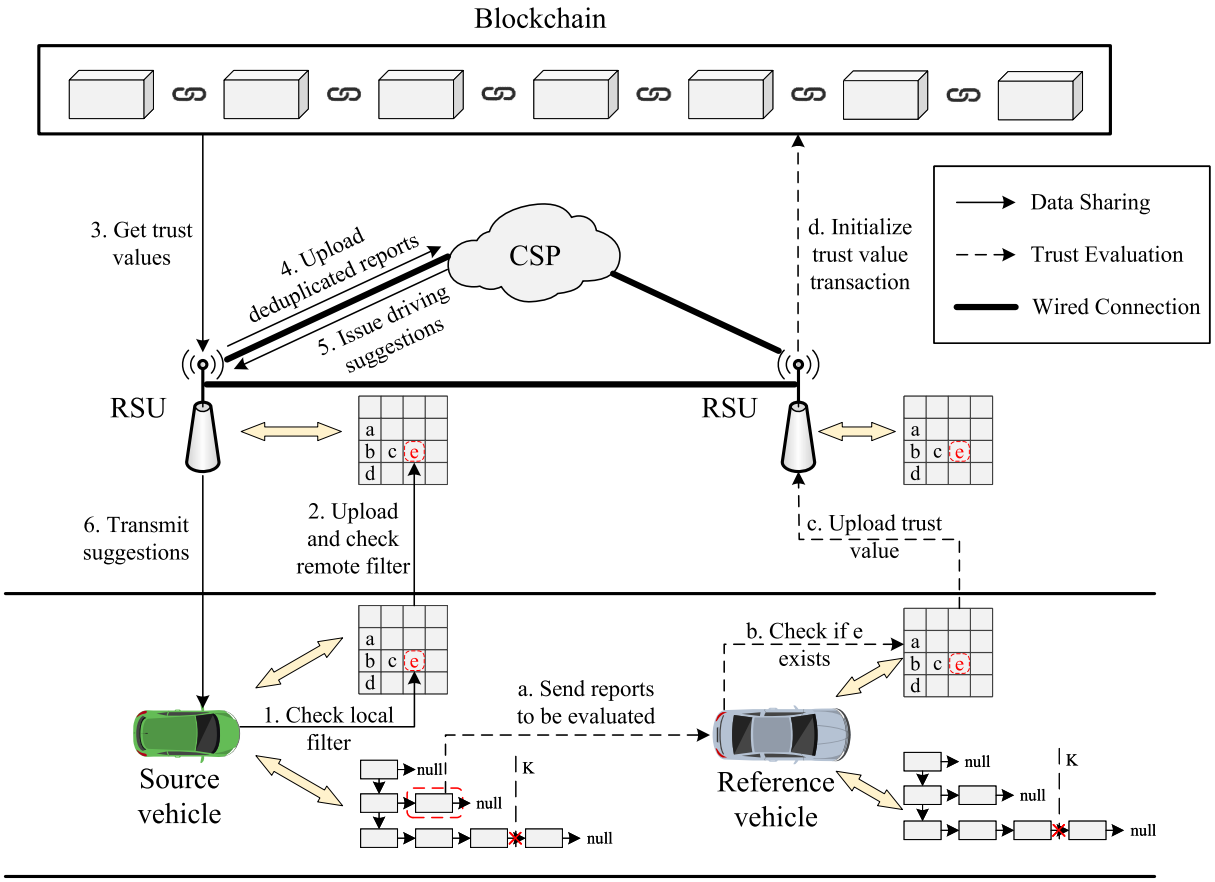


Fig. 3. System model of our TRUCON.

2) Road Side Unit (RSU): RSUs aggregate the data collected from the vehicles to the CSP. As a full node, RSU participates in the maintenance of the trust value of vehicle nodes in the blockchain. Besides, RSU delivers the necessary global traffic information and decision instructions to the vehicles in the same district.

3) Cloud Service Provider (CSP): CSP accepts and stores traffic information from RSUs which is collected from vehicles in this district. Since the vehicles themselves can only obtain the local traffic information, their driving decisions are usually optimal for itself rather than global situation. Therefore, CSP will perform massive computation tasks and provide delay-tolerant driving suggestions to improve traffic conditions.

4) Trust Authority (TA): TA is responsible for initializing necessary parameters and deploying blockchain system. It helps generate key pairs for every vehicle and RSU during the system initialization phase. Vehicles can encrypt the message to guarantee the security and privacy of the communications. He does not participate in the process of data sharing actually.

B. Threat Model

Vehicles and RSUs are the key participators in our proposed protocol. Attackers may try to compromise them for malicious purposes. Vehicles are more vulnerable to attackers because of limited computing ability and protection capability. On the contrary, RSUs have more powerful protection measures, which keep them away from attackers. The following threats are considered in TRUCON.

1) Vehicle Misbehavior: Vehicles According to their behaviors, vehicles can be divided into honest, lazy, and malicious ones. The misbehavior of vehicles in this paper mainly includes the following types.

- *Lazy sharing or evaluation:* Lazy vehicles are selfish, which means they don't share data or give scores while receiving data from others. It is not good for the stability and sustainability of the whole system.
- *Message spoofing:* A vehicle may report fake traffic information to RSUs and other vehicles due to hardware damages or malicious purposes.
- *Message inconsistency:* To escape regulation, malicious source vehicles may transmit correct messages to reference vehicles while sending fake messages to RSUs.
- *Giving unfair score:* When acting as reference vehicles, attackers may give a high score to fake message or low score to correct message.
- *Collusion attack:* Malicious vehicles may collude to give wrong data and scores while congregating in one district.

2) RSU Failure: RSUs are considered credible but unreliable in this paper. On the one hand, they are usually maintained by the governments and specialized enterprises, for which credibility can be guaranteed. However, they are unreliable because network interruption or system damage can lead to temporary RSUs service unavailable in a certain district.

TABLE I
NOTATIONS

Notation	Description
$area_k$	The k-th district on the road
RSU_k	Road side unit in $area_k$
CF_k	Cuckoo filter maintained by RSUs in $area_k$
α	Number of digits in area code
β	Number of digits in vehicle code
v_i	The i-th vehicle on the road
pk_{v_i}	Public key of v_i
sk_{v_i}	Private key of v_i
λ_i	Coordinate of v_i
R_i	Traffic data reports set held by v_i
Q_r	Trust degree of report r
T_i	Trust value of v_i
$H(x)$	Fingerprint algorithm used in cuckoo filter
$H_1(x)$	Hash function used in cuckoo filter
$S_{j\to i}$	Data quality score v_j gave to v_i
RT_i	Route table of v_i
$\Phi(R_i, R_j)$	Intersection between R_i and R_j

The blockchain system in this paper is jointly maintained by RSUs and vehicles. Since most of the nodes are honest, we can conclude that the blockchain is trustworthy. The deployed smart contract will record the trust value of each vehicle faithfully collecting from other vehicles. Besides, the blockchain system in our scheme is considered impossible to compromise, which means the trust value records on it can not be tampered with. Attacks on the IoV system itself such as the sybil attack are out of the scope of this paper.

V. DESIGN OF TRUCON

A. Overview of TRUCON

As mentioned in the previous part, we divide the whole region into m districts, among which the k-th one can be represented by $area_k$, $k \in \{1, 2, \ldots, m\}$. There exist several RSUs and vehicles in $area_k$. We denote certain vehicles in the target district as v_i, $i \in \{1, 2, \ldots, n\}$, its behavior and quality of data provided can be measured by trust value T_i. The following Table I shows all the notations used in this paper.

In order to relieve the pressure on the communication channel and ensure the credibility of shared data, we propose TRUCON, a blockchain-based trusted data sharing mechanism with congestion control in IoV. TRUCON mainly includes three phases: system initialization phase, data sharing phase, and trust evaluation phase. In the system initialization phase, TA deploys the blockchain system and smart contracts, after which involved entities invoke the contract to register on-chain. In the data sharing phase, we propose to utilize cuckoo filter on RSU to filter out the duplicate messages, which significantly reduces the communication overhead. Finally, source vehicles need to transmit their observed data to reference vehicles to ensure the credibility of sharing data. The trust values of vehicles will change according to scores from reference vehicles in the trust evaluation phase.

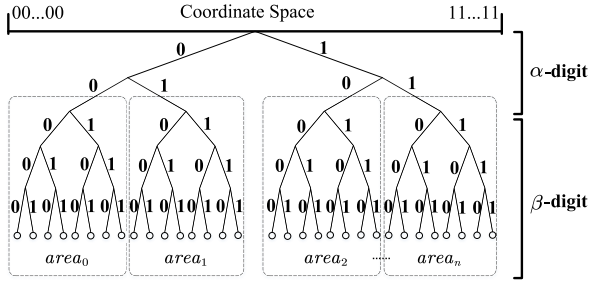


Fig. 4. Encoding method of vehicle coordinate λ_i.

B. System Initialization Phase

In this phase, the TA firstly deploys the blockchain system and smart contracts. The deployed blockchain is maintained by all of the RSUs and vehicles on the road, among which RSUs are full nodes while vehicles act as light nodes. It means that vehicles only need to store the block header information, which can be used to validate the authenticity of transactions using simplified payment verification (SPV) technology with limited storage capacity. After the blockchain system is deployed, vehicles can register by invoking smart contracts. The smart contract initializes a trust value T_i for every vehicle. TA follows the following steps to generate necessary system parameters.

1) Generate key pairs $< pk_{v_i}, sk_{v_i} >, i \in \{1, 2, \ldots, n\}$ for every vehicle v_i and $< pk_{RSU_j}, sk_{RSU_j} >, j \in \{1, 2, \ldots, m\}$ for every RSU.
2) Specify an asymmetric encryption algorithm to encrypt message msg, which can be denoted by $E_{sk_i}(msg)$.
3) Generate hash algorithms $H = \{0,1\}^* \to \{0,1\}^m$ to get the fingerprint and $H_1 = \{0,1\}^* \to \{0,1\}^n$ to calculate the first index of item x in filter CF_k. After that, broadcast it to all the registered entities.

To identify the vehicle with its real geographical location and trust value, we propose a coordinate encoding system based on Kademlia algorithm as shown in Fig. 4.

A full coordinate λ_i includes α-digit area code and β-digit vehicle code. Specifically, area code are used to encode districts on the road. Geographically close areas will be coded similarly. And every vehicle v_i will be assigned a unique β-digit binary number which is associated with its trust value. If v_i drives into another district, the α-digit area code will be updated accordingly while the vehicle code remains the same. In this way, λ_i contains geographical location characteristics and will not change for a period of time.

In addition to vehicles, RSUs also need to perform the necessary initialization process. Vehicles in $area_k$ maintain the same cuckoo filter CF_k by a wired connection. At first, every vehicle initializes an empty cuckoo filter, which is composed of buckets and one bucket can hold multiple entries. To further reduce communication overhead, vehicles will request CF_k from the nearest RSU and cache it locally.

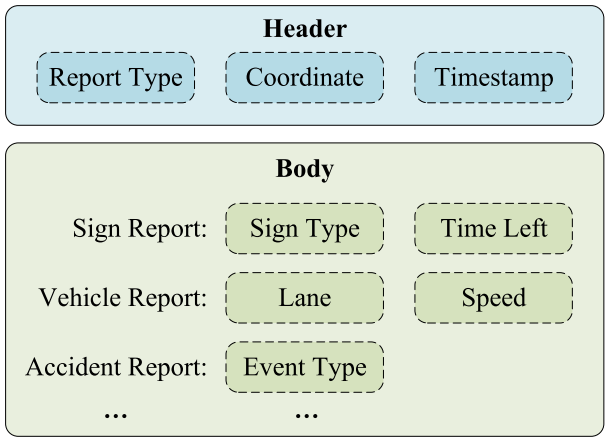


Fig. 5. Structure of data report.

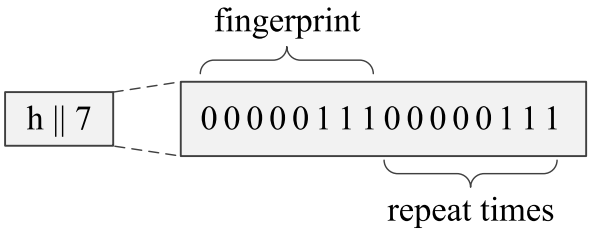


Fig. 6. Entry structure of CF_k.

C. Data Sharing Phase

As discussed earlier, our proposed scheme relies on the close cooperation between vehicles, RSUs, and CSP. Vehicles are equipped with multiple sensors, such as cameras, millimeter wave radar and lidar, etc. Upon vehicles joining the IoV, the sensors start observing traffic information on the road. Their perceived original data are then processed by the edge computing equipment using fusion methods mentioned in [31]. After performing a standardization process, the fused data can become a traffic event report. It should be noted that a scoring cycle in our scheme is divided into time periods Δt. So the hash value of traffic report will be the same if the perceived data does not change in Δt. The structure of the data report is shown in Fig. 5.

The standardized traffic data reports will be uploaded to RSUs so as to make better decisions to optimize the global traffic situation. However, chances are that vehicles in $area_k$ generate the same reports, which brings unnecessary communication overhead. To avoid this kind of situation, RSUs in $area_k$ will create a cuckoo filter CF_k to filter out the duplicate data. The entry structure of CF_k in our scheme is different from the original one in cuckoo filter. In addition to the fingerprint of item r, an entry also contains the number of times the corresponding item was uploaded, which can be shown in Fig. 6.

Vehicles and RSUs complete the data sharing phase through the following steps.

1) Source vehicles in $area_k$ send a sync request to the nearest RSU_k. After receiving the response, source vehicles cache CF_k locally to reduce communication overhead.
2) Every source vehicle v_i generates its traffic data report set R_i according to perception data from sensors.
3) Source vehicles check locally cached filter CF'_k to see if every r in report set R_i exists. If yes, discard r because it has already been uploaded. Otherwise, update CF'_k and continue with the following steps.
4) Source vehicle v_i initialize an encrypted request containing deduplicated report set R_r and its vehicle identity number (VIN) to the nearest RSU using its public key pk_{RSU_j}.

$$Req_i = E_{pk_{RSU_j}}(R_r \mid\mid VIN_i) \tag{1}$$

5) RSU_k decrypts messages from vehicles and checks if every r in R_i already exists in local CF_k. If yes, update the repeat time t_r only. Otherwise, insert report r into filter CF_k and set the repeat time t_r as one. The entry of r can be represented by Equation 2.

$$entry(r) = < H(r) \mid\mid t_r > \tag{2}$$

6) RSU_k uploads deduplicated reports to CSP. Chances are that two conflicting reports r and r' exist at the same time. To decide which is credible, we need to calculate the trust degree Q_r of r.

$$Q_r = \frac{t_r}{t_r + t_{r'}} \tag{3}$$

If $Q_r > 0.5$, RSU discards r' and uploads r. If $Q_r < 0.5$, do the opposite. Otherwise, invoke the smart contract to get trust values of v_i and v_j, who upload report r and r' respectively. Then, decide whose report to accept based on trust values.
7) CSP generates and issues driving suggestions based on reports. RSUs are responsible for transmitting them to vehicles.

Algorithm 1 shows the specific process of the above steps.

D. Trust Evaluation Phase

Since the vehicles in our scheme are not trustworthy, they may transmit fake messages to RSUs due to hardware damages or malicious purposes. To make vehicles behave honestly and reduce communication overheads, we propose a unique trust management mechanism. RSUs will initiate the trust evaluation process occasionally. As mentioned in Section IV, vehicles are divided into source vehicles and reference vehicles in our scheme. Source vehicle v_i needs to transmit report set R_i uploaded in the last time period Δt to other vehicles. The whole data evaluation process is composed of two major parts. At first, source vehicle v_i needs to decide which reference vehicles to forward these reports to. Secondly, reference vehicle v_j gives a score to v_i and invoke smart contracts to update v_i's trust value T_i.

Specifically, v_i completes the data forwarding phase using Kademlia algorithm according to the following steps:

1) Source vehicle v_i sends a request to the nearest RSU_k to fetch the active reference vehicles list information. The list contains VIN_j and their coordinate λ_j.

Algorithm 1 Deduplicated Traffic Reports Uploading

Input: Traffic report set R_i, Trust values T_i
Output: Cuckoo filter CF_k

for $r \in R_i,\ i \in \{1, 2, \ldots, n\}$ **do**
  $f = fingerprint(r)$
  $i_1 = hash(r),\ i_2 = i_1 \oplus hash(f)$
  if local filter CF'_k contains r **then**
    update repeat times t_r in CF'_k
    continue
  else
    Add $< H(r) \mid\mid 1 >$ to CF'_k
    Send $Req_i = E_{pk_{RSU_j}}(r \mid\mid VIN_i)$ to RSU_j
  end if
  RSU_j decrypts Req_i to get r
  if CF_k contains f **then**
    update $< H(r) \mid\mid t_r >$ to $< H(r) \mid\mid t_r + 1 >$
  else if $bucket[i_1]$ or $bucket[i_2]$ has empty entry **then**
    select a entry to insert $< H(r) \mid\mid 1 >$
  else
    return Failure
  end if
end for
return CF_k

Algorithm 2 Traffic Reports Forwarding

Input: Vehicle coordinate λ_j
Output: Traffic report set R_i

for $v_j,\ j \in \{1, 2, \ldots, m\}$ **do**
  $d(i, j) = \lambda_i \oplus \lambda_j$
  if $d(i, j) \geq d$ **then**
    v_i discard v_j in RT_i and **continue**
  end if
  find index x satisfying $d(i, j) \in [2^x, 2^{x+1})$
  if k-bucket[x] is not full **then**
    insert v_j to k-bucket[x]
  else
    remove invalid node in k-bucket[x]
    insert v_j to k-bucket[x]
  end if
  v_i transmits R_i to v_j in RT_i
end for
return R_i

2) For every reference vehicle $v_j,\ j \in \{1, 2, \ldots, m\}$, v_i calculates their XOR distance $d(i, j)$ respectively, which can be represented by Equation 4.

$$d(i, j) = \lambda_i \oplus \lambda_j \tag{4}$$

3) v_i constructs its routing table RT_i using distances $d(i, j),\ j \in \{1, 2, \ldots, m\}$. RT_i is composed of multiple k-buckets. The i-th k-bucket holds reference vehicles whose distance between v_i varies from 2^i to 2^{i+1}. Every k-bucket contains at most k vehicle nodes. The structure of RT_i is shown in Fig. 2.
4) Because the closer two vehicles are, the more likely they generate the same traffic reports. We set a distance threshold d for the routing tables. Then, the source vehicles forward traffic reports to reference vehicles with a distance within than d.

Algorithm 2 elaborates the details of the above process.

Upon receiving the traffic report set R_i from source vehicle v_i, reference vehicle v_j will start the trust evaluation process as the following steps.

1) Reference vehicle v_j decrypts messages msg using its private key sk_{v_j} to get R_i.

$$R_i = E^{-1}_{sk_{v_j}}(msg) \tag{5}$$

2) R_j represents the traffic report set held by v_j. There is a partial overlap between R_i and R_j, which is denoted as $\Phi(R_i, R_j)$. For every report in $\Phi(R_i, R_j)$, v_j will check its local filter CF_k to see if it exists. If not, v_j will report a malicious behavior to RSU_k. Finally, v_j will compare the existing reports with the local data to see if they are correct.

$$\Phi(R_i, R_j) = R_i \cap R_j \tag{6}$$

3) Then v_j calculates score $S_{j\to i}$ based on the data quality of v_i after comparing with its locally observed data.

$$S_{j\to i} = \frac{N(c)}{N(c) + N(\overline{c})} \tag{7}$$

In the above Equation 7, $N(c)$ represents the correct number of data reports that v_j thinks in $\Phi(R_i, R_j)$. Similarly, $N(\overline{c})$ denotes the number of reports which is not correct.
4) Reference vehicle v_j invokes the smart contract to update the trust value of v_i. Since v_j is a light node in blockchain, it requests missing data from the nearest RSU_k when needed. Besides, v_j is not reliable itself, the credibility of $S_{j\to i}$ depends on the trust value of v_j, which is denoted by T_j.
5) After all the reference vehicles give their scores towards R_i, the average score of v_i at this round can be represented as:

$$S_i = \frac{\sum_{j=1}^{m} T_j \cdot S_{j\to i}}{m} \tag{8}$$

6) The average score S_i can affect the trust value T_i eventually. To keep trust values under control when they fluctuate in order to keep the system stable, we adopt a reasonable value changing manner.

$$T'_i = \begin{cases} T_i + (S_i - T_i)^{\rho} & S_i \geq T_i \\ T_i - (T_i - S_i)^{\sigma} & S_i < T_i \end{cases} \tag{9}$$

In the above equations 9, $\rho \in (0, 1)$ while $\sigma \in (1, +\infty)$. When S_i is higher than the original T_i, trust value T_i should increase with the difference between them and show a trend of fast first and then slow. When S_i is lower than the original T_i, the declining trust value T_i should show an opposite trend which means changing slow first and then fast.

7) Since vehicles act as light nodes in the blockchain, they relies on RSUs to initialize trust value transactions. If vehicle v_i shows laziness when sharing data, RSUs will decrease its trust value T_i in a small time slice. At the end of a scoring cycle, T_i will suffer a sudden change according to the scores from reference vehicles. When malicious behavior such as message inconsistency detected, T_i will shrink as Equation 10, where μ represents the malicious behavior times of v_i .

$$T_i' = T_i \cdot (1-\varepsilon)^{\mu} \tag{10}$$

The details mentioned above can refer to Algorithm 3.

Algorithm 3 Trust Evaluation

Input: Report set R_i and R_j, Original trust value T_i
Output: Trust value T_i'

for v_j, $j \in \{1, 2, \ldots, m\}$ **do**
  $\Phi(R_i, R_j) = R_i \cap R_j$
  for report $r \in \Phi(R_i, R_j)$ **do**
    if $H(r)$ is not contained in CF_k **then**
      update trust value $T_i' = T_i \cdot (1-\varepsilon)^{\mu}$
    else if v_j thinks r is correct **then**
      update $N(c) = N(c) + 1$
    else
      update $N(\overline{c}) = N(\overline{c}) + 1$
    end if
  end for
  calculate $S_{j\to i} = N(c) \ / \ (N(c) + N(\overline{c}))$
end for
get average score $S_i = \sum_{j=1}^{m} T_j \cdot S_{j\to i} \ / \ m$
if $S_i \geq T_i$ **then**
  update trust value $T_i' = T_i + (S_i - T_i)^{\rho}$
else
  update trust value $T_i' = T_i - (T_i - S_i)^{\sigma}$
end if
return T_i'

VI. Security Analysis

A. Defense Against Vehicle Misbehavior

1) Lazy Sharing or Evaluating: Lazy vehicles are selfish in the data sharing process. They only receive traffic information from RSUs and other vehicles but do not share data and give scores actively, which can result in the system failing to operate continuously. To prevent this from happening, we propose to automatically decay the trust values of vehicles that do not share data for a period of time. This process is executed through RSU_k. Especially, offline vehicles are out of the consideration of lazy sharing or evaluating.

2) Message Spoofing: Malicious vehicles may share fake traffic reports with other vehicles, which may cause CSP to make wrong decisions. To reduce the impact of message spoofing attacks, we set a data judgment process in the data uploading phase. Because most source vehicles behave honestly and upload correct traffic reports, $RSUk$ will calculate Q_r according to Equation 3. More than half of the source vehicles behaving honestly can ensure that RSU_k pick out the correct information.

3) Message Inconsistency: Malicious vehicles may upload fake traffic reports to RSU_k while sharing correct information to reference vehicles. In that way, it may affect traffic safety without penalty. However, because the check is suddenly initiated by RSUs and all reference vehicles contain the filter CF_k inside. Reference vehicles will check CF_k to see if the fingerprints of those traffic reports are recorded, which identifies message inconsistencies in time.

4) Giving Unfair Score: Reference vehicles may also give an unfair score to source vehicles for profits. To weaken the effects brought by unfair scores, we adopt the weighted average value of all scores from reference vehicles. The weight of every vehicle come from its own trust values. When the times of giving an unfair score reach a threshold, it is identified as malicious and triggers the trust value decay process according to Equation 10.

5) Collusion Attack: When sharing data, malicious vehicles may congregate in one district, colluding to make up an absolute majority in areas with few vehicles. If the number of vehicles in this district is less than a lower threshold and traffic reports conflict occurs, RSU_k will check each traffic data report with its local observed traffic data to defend against collusion attacks.

B. Defense Against RSU Failure

Since RSUs are maintained by the government or specialized enterprise, it is considered trustworthy. However, it is possible that RSUs break down due to temporary hardware failure. For example, if RSU_k in $area_k$ is out of service temporarily, vehicles in that area can require the same service from another RSU. According to the scope of the failure, RSU failure can be divided into individual and regional one.

1) Individual Failure: Because RSUs in $area_k$ share the storage space and filter CF_k, vehicles v_i can seamlessly switch to another RSU in $area_k$. v_i do not need to update system parameters such as filter CF_k, area code of *VIN*, and so on.

2) Regional Failure: When a regional failure occurs, v_i need to request service from RSU in another area. First, RSU will reply to it with multiple parameters such as public key of RSU, area code of the target area, and filter CF_{k+1}. Then v_i updates its local filter to CF_{k+1} and the switch process is over.

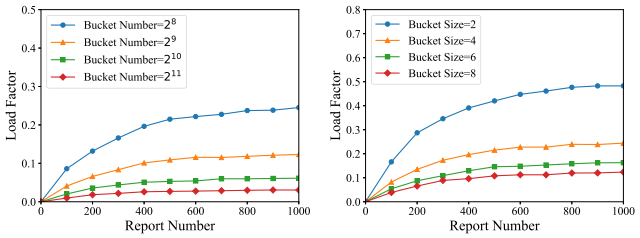
VII. Performance Evaluation

A. Experiment Setup

In order to evaluate the effectiveness and performance of TRUCON, we implement a prototype system on PC with an Intel Core CPU i7-10700F and 16 GB RAM. Specially, we initialize 1000 vehicles with unique $VINs$ evenly distributed in 10 areas. In the off-chain part, we implement Kademlia algorithm in python and modify the imported *cuckoopy* package to test the communication overhead. As for the on-chain part, we build a private blockchain using Ethereum client *Geth* 1.10.4 and develop smart contracts using the *Truffle* framework. The RSUs are responsible for generating blocks

TABLE II
SYSTEM PARAMETERS

Parameter	Description	Value
α	Digit of area code	6
β	Digit of vehicle code	10
BS	Bucket size of filter CF_k	4
BN	Bucket number of filter CF_k	256
FS	Fingerprint size of traffic report	8 bits
k	k-bucket capacity	20
d	Distance threshold in k-bucket	2^{16}
Δt	Block generating interval	3s
ρ	Trust growth factor	0.5
σ	Trust decay factor	2
ε	Misbehavior decay factor	0.2



(a) Impact of bucket number (b) Impact of bucket size

Fig. 7. Impact of filter parameters on load factor.

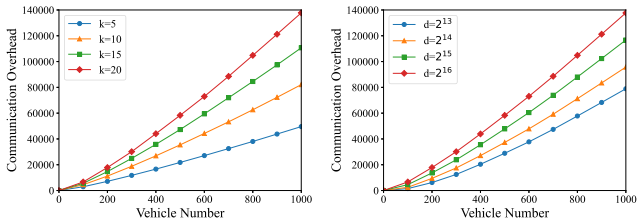
and reaching consensus using Clique. Table II lists the system parameters used in our experiment.

B. Off-Chain Part

1) Impact of System Parameters: To achieve the best effect of our data sharing mechanism, we conduct a detailed experiment on the impact of system parameters. In the data sharing phase, the filter CF_k is composed of two major parameters: bucket number and bucket size. Both of them can affect the load factor of filter CF_k, which reflects the probability of entry displacement. The followings are our experiment results.

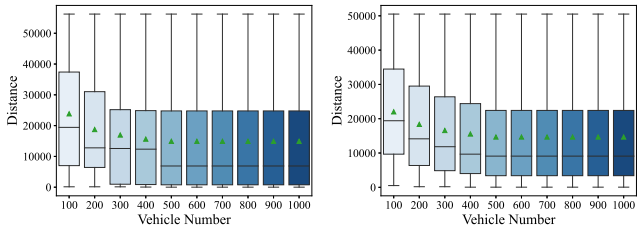
As shown in Fig. 7, we can see that as bucket number and bucket size grow, the filter load factor increases continuously. The growing rate increases slower when the filter CFk fills up. It is because more and more repeated fingerprints are encountered during this process. On the one hand, we should keep low load factor as soon as possible to reduce unnecessary displacements. On the other hand, we should make CF_k as small as possible to reduce the synchronization burden. Considering the study of aforementioned experimental findings, we select bucket size to be 4 and bucket number to be 256.

Similarly, we conduct the corresponding experiments during the trust evaluation phase to study the impact of Kademlia parameters on communication overhead. As shown in Fig. 8, we use the number of sending traffic reports to represent the communication overhead of the network channel. Communication overhead increases with increasing values of k and d. We adopt k as 20 and d as 2^{16} in the following experiments.



(a) Impact of k value (b) Impact of d value

Fig. 8. Impact of Kademlia parameters on communication overhead.



(a) Vehicles divided into 10 areas (b) Vehicles divided into 50 areas

Fig. 9. Distribution of vehicle distances.

2) Distribution of Vehicle Distances: In TRUCON, we encode every vehicle using the combination of area code and vehicle code. So the granularity of area division has a great impact on the distance between vehicles. We initialize 500 vehicles evenly distributed in each area to explore their relationships in this part. Fig. 9a shows the distribution of vehicle distances when we divide the vehicles into 10 areas while Fig. 9b divides them into 50 areas. Obviously, the distance decreases as more vehicles are present before stabilizing. It is because the greater the density of vehicles, the smaller distances between vehicles will be. Besides, a finer-grained area division leads to an increase in overall distances. When the number of divided areas changes from 10 to 50, the overall distances become greater because there are fewer vehicles in a single district.

3) Communication Overhead: To evaluate the effectiveness of our scheme, we conduct a comparison experiment with another two typical data sharing schemes. As shown in Fig. 10, TRUCON performs better than reference [12] but worse than CADC [26]. But as introduced earlier, CADC is only a data collection method without trust management. It means there is a lack of credibility in the data shared by CADC. Reference [12] proposes a blockchain-based decentralized trust management mechanism. Vehicles broadcast their observed data directly to others, among which many vehicles do not have the ability to judge the correctness of the data reports. TRUCON implements adaptive congestion control under the premise of achieving trusted data sharing at the cost of certain computing costs.

C. On-Chain Part

1) Calculation of Trust Value: One of the most important function of blockchain system in our scheme is to record the trust values of all the vehicles. There are three different

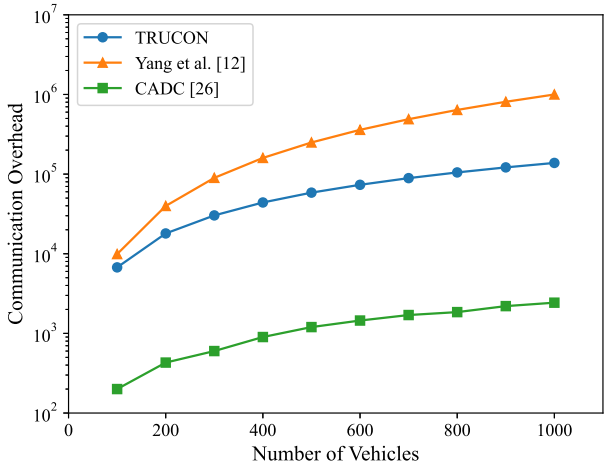


Fig. 10. Communication overhead comparison in data sharing.

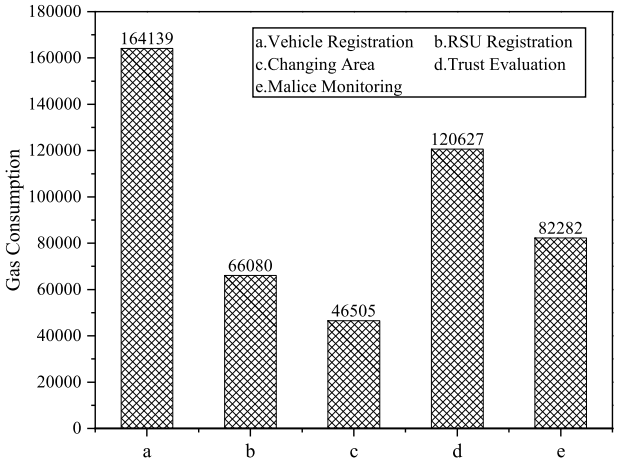


Fig. 12. Gas consumption of each operation.

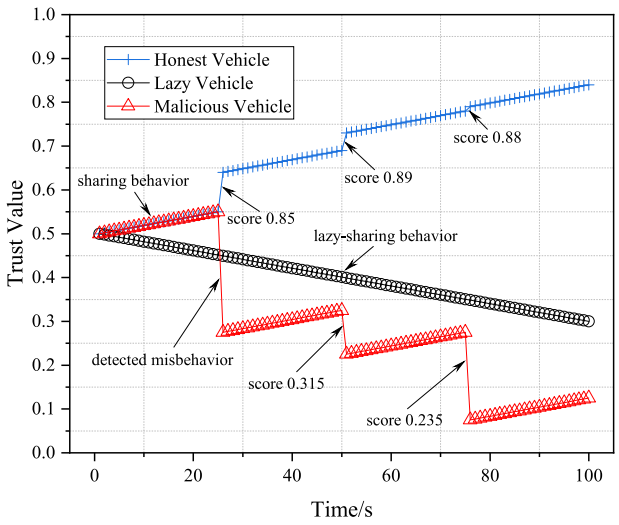


Fig. 11. Calculation of trust value.

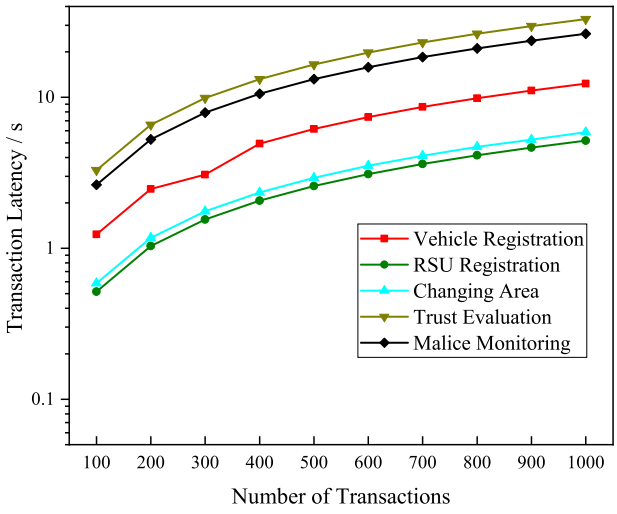


Fig. 13. Time latency of each operation.

types of vehicles in our system, which are honest vehicle, lazy vehicle and malicious vehicle. Every vehicle is given an initial trust value according to current average level of all the vehicles, which is set as 0.5 in our experiment. Honest vehicles and malicious vehicles are willing to share the traffic information they observe to increase their trust values, while lazy vehicles do the opposite. The trust value of vehicle will increase if it gets a score higher than its trust value. In contrast, its trust value will decrease if it gets a lower score. So as we can see from Fig. 11, honest vehicles usually get a higher score during a scoring round while malicious vehicles usually get a lower score. The trust value of lazy vehicles will keep decreasing because of their lazy-sharing behavior.

2) Gas Consumption: In Ethereum, transaction initiator needs to pay the miners for computation costs. And the total fee is determined by specified gas price and limit. As shown in Fig. 12, it takes 5 operations to realize our scheme on the blockchain. Among them, vehicle registration is the most consuming operation, which takes more than 164,000 units of gas. This is because we need to specify the value of a series of parameters to initialize a vehicle. Operation (b) takes less than 77,000 units of gas, as the number of parameters needed in the RSU registration process is much less than vehicle registration. The operation of changing area consumes the minimal gas, which is less than 50,000 units. It avoids complex operations for vehicles to changing areas. Finally, trust evaluation and malice monitoring consumes about 120,000 and 82,000 units of gas respectively. Obviously, the operation cost of the on-chain part is acceptable and affordable in practice.

3) Transaction Latency: Transaction latency reflects the speed of transaction processing in blockchain. We initiate 1,000 transactions in the transaction pool and start mining at the same time to evaluate the latency of the different number of transactions for every operation. Fig. 13 shows the delay required by different operations on the blockchain under different transaction numbers. As expected, the operation of trust evaluation spends the most time handling these transactions, which takes nearly 33s. When the number of transactions is 100, the time needed is 3.3s. It is nearly the block time we preset before. And the latency for the operation of monitoring

another vehicle's behavior follows closely, which is about 26s for handing 1,000 transactions. The operations with the least delay are changing areas and RSU registration. They spend 0.59s and 0.52s respectively for 100 transactions. Therefore, it is convenient for vehicles to move to another district.

VIII. Conclusion

In this paper, we propose TRUCON, a blockchain-based trusted data sharing mechanism with congestion control to solve the efficiency and trust problems existing in the IoV. Our scheme is composed of two phases: the data sharing phase and trust evaluation phase. We propose a traffic data deduplication and discrimination method based on cuckoo filter in the data sharing phase. By checking local filter entries in RSUs and vehicles, our scheme can prevent massive redundant data from being shared. The repeat times of records can be used to judge the credibility of certain data reports when conflict happens. Trust evaluation can be divided into two parts actually. Firstly, we introduce the Kademlia algorithm into our scheme to achieve traffic data forwarding. Every source vehicle can construct its own routing table based on its XOR distances between reference vehicles. So we can control the channel congestion situation by changing values of the bucket capacity k and distance threshold d. Secondly, reference vehicles invoke the smart contracts to give scores to source vehicle. Experiment result shows that our scheme saves 31% of communication overhead compared with existing trust management solution when there are 100 vehicles. And if there are 1000 vehicles, it saves about 86%.

The shortcoming of our current work is that applicable scenarios are limited and we did not take privacy issues into consideration. The effect of congestion control is limited when there are few vehicles. In the future, we will keep trying to reduce the communication overhead. Besides, we will also work to protect the security of private information of vehicles such as their real identity and location.

References

[1] *Global Status Report on Road Safety 2018*, World Health Organization, Geneva, Switzerland, 2018.

[2] A. Hbaieb, S. Ayed, and L. Chaari, "A survey of trust management in the Internet of Vehicles," *Comput. Netw.*, vol. 203, Feb. 2022, Art. no. 108558.

[3] Z. Xu, W. Liang, K.-C. Li, J. Xu, and H. Jin, "A blockchain-based roadside unit-assisted authentication and key agreement protocol for Internet of Vehicles," *J. Parallel Distrib. Comput.*, vol. 149, pp. 29–39, Mar. 2021.

[4] S. Duan et al., "Multitype highway mobility analytics for efficient learning model design: A case of station traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19484–19496, Oct. 2022.

[5] S. Abbes and S. Rekhis, "A blockchain-based solution for reputation management in IoV," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2021, pp. 1129–1134.

[6] Z. Su, Y. Wang, Q. Xu, and N. Zhang, "LVBS: Lightweight vehicular blockchain for secure data sharing in disaster rescue," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 19–32, Jan. 2022.

[7] S. Nakamoto. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[8] Y. Xu, Z. Liu, C. Zhang, J. Ren, Y. Zhang, and X. Shen, "Blockchain-based trustworthy energy dispatching approach for high renewable energy penetrated power systems," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 10036–10047, Jun. 2022.

[9] C. Feng, B. Liu, K. Yu, S. K. Goudos, and S. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3582–3592, May 2022.

[10] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, "A blockchain-based multi-cloud storage data auditing scheme to locate faults," *IEEE Trans. Cloud Comput.*, early access, Feb. 8, 2021, doi: 10.1109/TCC.2021.3057771.

[11] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 289–300, Mar. 2020.

[12] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.

[13] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Proc. Int. Workshop Peer Peer Syst.* Cambridge, MA, USA: Springer, 2002, pp. 53–65.

[14] Q. Li, A. Malip, K. M. Martin, S.-L. Ng, and J. Zhang, "A reputation-based announcement scheme for VANETs," *IEEE Trans. Veh. Technol.*, vol. 61, no. 9, pp. 4095–4108, Nov. 2012.

[15] M. E. Mahmoud and X. Shen, "An integrated stimulation and punishment mechanism for thwarting packet dropping attack in multihop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 8, pp. 3947–3962, Oct. 2011.

[16] C. Lai, K. Zhang, N. Cheng, H. Li, and X. Shen, "SIRC: A secure incentive scheme for reliable cooperative downloading in highway VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1559–1574, Jun. 2017.

[17] I. García-Magariño, S. Sendra, R. Lacuesta, and J. Lloret, "Security in vehicles with IoT by prioritization rules, vehicle certificates, and trust management," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 5927–5934, Aug. 2019.

[18] C. Boudagdigue, A. Benslimane, A. Kobbane, and M. Elmachkour, "A distributed advanced analytical trust model for IoT," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[19] Z. Liu et al., "BTMPP: Balancing trust management and privacy preservation for emergency message dissemination in vehicular networks," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5386–5407, Apr. 2021.

[20] P. K. Singh, R. Singh, S. K. Nandi, K. Z. Ghafoor, D. B. Rawat, and S. Nandi, "Blockchain-based adaptive trust management in Internet of Vehicles using smart contract," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3616–3630, Jun. 2020.

[21] X. Liu, H. Huang, F. Xiao, and Z. Ma, "A blockchain-based trust management with conditional privacy-preserving announcement scheme for VANETs," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4101–4112, May 2020.

[22] S. Liu, J. Yu, X. Deng, and S. Wan, "FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1616–1629, Feb. 2022.

[23] A. K. Sangaiah, J. S. Ramamoorthi, J. J. P. C. Rodrigues, M. A. Rahman, G. Muhammad, and M. Alrashoud, "LACCVoV: Linear adaptive congestion control with optimization of data dissemination model in vehicle-to-vehicle communication," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5319–5328, Aug. 2021.

[24] G. E. M. Zhioua, N. Tabbane, H. Labiod, and S. Tabbane, "A fuzzy multi-metric QoS-balancing gateway selection algorithm in a clustered VANET to LTE advanced hybrid cellular network," *IEEE Trans. Veh. Technol.*, vol. 64, no. 2, pp. 804–817, Feb. 2015.

[25] G. Luo et al., "Software-defined cooperative data sharing in edge computing assisted 5G-VANET," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 1212–1229, Mar. 2021.

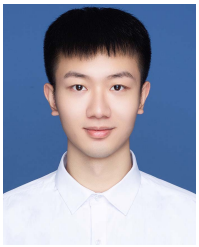
[26] Y. Zhuang et al., "Data collection with accuracy-aware congestion control in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 5, pp. 1068–1082, May 2019.

[27] J. Wang et al., "Dynamic clustering and cooperative scheduling for vehicle-to-vehicle communication in bidirectional road scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1913–1924, Jun. 2018.

[28] J. Sun, H. Xiong, S. Zhang, X. Liu, J. Yuan, and R. H. Deng, "A secure flexible and tampering-resistant data sharing system for vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12938–12950, Nov. 2020.

[29] 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.

[30] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo filter: Practically better than Bloom," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2014, pp. 75–88.
[31] Y. Cui et al., "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 722–739, Feb. 2021.



Mingyang Yuan (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering and automation from Central South University, Changsha, China, in 2020, where he is currently pursuing the master's degree with the School of Computer Science and Engineering. His research interests include blockchain, trusted computing, and information security.

Yang Xu (Member, IEEE) received the Ph.D. degree in computer science and technology from Central South University, China, in 2019. From 2015 to 2017, he was a Visiting Scholar at the Department of Computer Science and Engineering, Texas A&M University, USA. He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha. He also works with the Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, Zhejiang University, Hangzhou. He has published over 50 articles in international journals and conferences, including the IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. His research interests include distributed computing, cloud computing, blockchain, artificial intelligence, and trustworthy/privacy computing. He is a member of the Blockchain Technical Committee of China Computer Federation (CCF) and the China Society for Industrial and Applied Mathematics (CSIAM), and a member of ACM. He was the Awardee of the Best Paper Award of IEEE International Conference on Internet of People (IoP 2018).

Cheng Zhang (Student Member, IEEE) received the B.S. degree in computer science and technology from the Shenyang University of Technology, China, in 2017, and the M.S. degree from Central South University, China, in 2021. He is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, China. He also works at the Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, Zhejiang University, China. His research interests include network security and blockchain.

Yunlin Tan received the B.S. degree in communication engineering from Hunan Normal University, Hunan, China, in 2021. He is currently pursuing the M.Sc. degree in information and communication engineering with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include wireless networks, blockchain technique, and cloud computing.

Yichuan Wang (Member, IEEE) received the Ph.D. degree in computer system architecture from Xidian University, China, in 2014. He is currently an Associate Professor with the Xi'an University of Technology and with the Shaanxi Key Laboratory of Network Computing and Security Technology. His research interests include cloud computing and networks security. He is an ACM Member and a CCF Member.

Ju Ren (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Central South University, China, in 2009, 2012, and 2016, respectively.

He is currently an Associate Professor with the Department of Computer Science and Technology, Tsinghua University, China. His research interests include the Internet of Things, edge computing, operating systems, and as well as security and privacy. He received many Best Paper Awards from IEEE flagship conferences, including IEEE ICC 2019 and IEEE HPCC 2019, the IEEE TCSC Early Career Researcher Award in 2019, and the IEEE ComSoc Asia-Pacific Best Young Researcher Award in 2021. He also served as the General Co-Chair for IEEE BigDataSE 2020, the TPC Co-Chair for IEEE BigDataSE 2019, a Poster Co-Chair for IEEE MASS 2018, and a Track Co-Chair for IEEE/CIC ICCC 2019, IEEE I-SPAN 2018, and VTC 2017 Fall. He also serves as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and *Peer-to-Peer Networking and Applications*. He was recognized as a Highly Cited Researcher by Clarivate from 2020 to 2022.

Yaoxue Zhang (Senior Member, IEEE) received the B.Sc. degree from the Northwest Institute of Telecommunication Engineering, China, in 1982, and the Ph.D. degree in computer networking from Tohoku University, Japan, in 1989. He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He has published over 200 papers. His research interests include computer networking, operating systems, and transparent computing. He is a fellow of the Chinese Academy of Engineering. He is the Editor-in-Chief of *Chinese Journal of Electronics*.