$\mathcal{R}$-manager: Consortium Blockchain-based Vehicle Reputation Management for High-quality Reports in Traffic-oriented Crowdsourcing

Enze Yu, Yuwei Xu, *Member, IEEE,* Lin Gao, Jie Cao, *Student Member, IEEE,* Qiao Xiang, *Member, IEEE,* Liang He, *Senior Member, IEEE* .

***Abstract*—The rise of 5G communication technology brings opportunities to traffic-oriented crowdsourcing in the Internet of Vehicles (IoV). As vehicles act as workers in many traffic-oriented crowdsourcing systems, there's a growing focus on how to get high-quality reports when some vehicles may be malicious or selfish. Some researchers assign reputation values to vehicles and select high-reputation vehicles to complete tasks, but these efforts still have shortcomings. In model design, guaranteeing the objectivity of reputation calculation, suppressing malicious and selfish behaviors, and maximizing the utilities of honest vehicles and service providers should be considered simultaneously. In system construction, reputation verification service in multi-party intelligent transportation scenarios and reputation management system performance should be satisfied. In this paper, we propose $\mathcal{R}$-manager, a consortium blockchain-based vehicle reputation management scheme. Firstly, we design a reputation model that uses confirmed results to update reputation and suppress malicious and selfish behaviors based on reputation deposit forfeit and a tax mechanism. Besides, the model maximizes the utilities of crowdsourcing participants by reaching the game equilibrium. Secondly, we design a reputation management system based on a consortium blockchain to abstract management activities as transactions and verify them by executing smart contracts, which meet multi-party management and reputation verification needs. Moreover, we design a half-committee endorsement strategy to improve system performance. Finally, our model is verified by simulation, and it can better suppress malicious and selfish behaviors compared with three different reputation models. We implement a prototype system and evaluate its performance. $\mathcal{R}$-manager outperforms two state-of-the-art blockchain-based schemes.**

***Index Terms*—Reputation management, 5G Internet of Vehicles, Consortium blockchain, Crowdsourcing.**

I. Introduction

The Internet of Vehicles (IoV) has become the cornerstone of the intelligent transportation system with the rapid development of 5G communication technology in recent years [1]. As an indispensable mode of information interaction in IoV, crowdsourcing can utilize the idle resources of vehicles to accomplish various tasks. Traffic-oriented crowdsourcing can support many applications such as road status detection [2], electric vehicle charging query [3], and arrival-departure time plan [4]. Its typical scene is that a service provider (SP) releases task information to nearby vehicles, and they choose to take charge of a task and report results. The task results reported from vehicles directly influence crowdsourcing applications. Since the vehicles are manned, they may behave maliciously or selfishly. Malicious vehicles may offer wrong task results to interfere with crowdsourcing applications, and selfish vehicles may neglect all the tasks for the purpose of reducing overhead. Obviously, high-quality reports from vehicles are the basis for supporting crowdsourcing applications. High quality encompasses two levels of meaning. On the one hand, vehicles actively participate in the reporting. On the other hand, the vehicle earnestly completes the task and reports the precise task result.

Cryptography theory [5] [6] can only guarantee the confidentiality and integrity of reports in the transmission process but cannot confirm the authenticity of report contents. To encourage vehicles actively participate in crowdsourcing tasks and respond to high-quality reports, some scholars [7] [8] attempt to assign reputation values to vehicles and recruit high-reputation vehicles to execute tasks. Therefore, how to manage the reputation of vehicles is very important. For the traffic-oriented crowdsourcing scenario, a complete reputation management solution should include two parts: the design of the reputation model and the construction of the reputation management system. The first part defines how to calculate reputation values and the second focus on how to apply the reputation model in a practical system.

Many methods such as game theory [9] and deep learning [10] are utilized to build reputation models. Three points should be considered when designing a reputation model in a traffic-oriented crowdsourcing application. First, the reputation calculation basis should be confirmed. Most models rely on subjective judgments [11] and feedback [12] from task initiators and neighboring vehicles to calculate reputation values. However, the above reputation calculation basis is unconfirmed and may be forged by malicious vehicles. Second, malicious and selfish behaviors should be suppressed. Existing models [10] [12] focus on incentivizing vehicles to participate in crowdsourcing while ignoring malicious behaviors to be suppressed. In addition, the incentive mechanism cannot

This work was supported in part by the National Key R&D Program of China (No. 2020YFB1005500), in part by the National Natural Science Foundation of China (No. 61702288), in part by the Fundamental Research Funds for the Central Universities, Southeast university. (Corresponding Author: Yuwei Xu)

E. Yu, Y. Xu, L. Gao, and J. Cao are with the School of cyber science and engineering, Southeast University, Nanjing 211189, China (e-mail: enzey@seu.edu.cn, xuyw@seu.edu.cn, gao_lin@seu.edu.cn, jie_cao@seu.edu.cn)

Q. Xiang is with the School of informatics, Xiamen University, Xiamen 361000, China. (qiaoxiang@xmu.edu.cn).

L. He is with the School of Computing, University of Nebraska-Lincoln, Nebraska 68512, USA. (lhe@unl.edu).

kick out vehicles that are always selfish and unwilling to participate in crowdsourcing. Third, the utilities of honest vehicles and service providers should be maximized when they participate in crowdsourcing. Existing studies [13] [14] ignore maximizing utility and lack sufficient motivation to engage vehicles in tasks. In general, current reputation models cannot simultaneously achieve three goals: using a confirmed calculation basis, suppressing malicious and selfish behaviors, and maximizing the utilities of honest vehicles and service providers.

The reputation management system can be constructed in centralization or decentralization ways. The centralized reputation management system [15] [16] not only has performance bottlenecks and single point of failure risks but also cannot meet multi-party reputation management needs. Decentralized reputation management systems can be designed in different architectures [17] [18]. These methods rely on vehicles or roadside units to manage reputation. Although the system scalability is improved, these distributed nodes neither maintain a unified vehicle reputation record nor provide publicly verifiable services. Due to the advantages of immutability and public traceability, blockchain technology is introduced by researchers to provide reputation verification services [19] [20]. These blockchain systems rely on multi-party consensus to verify reputation, but their consensus mechanism is based on Proof of Work (PoW), which results in the gap between system performance and practical reputation management needs.

In this paper, we propose a consortium blockchain-based vehicle reputation management scheme called $\mathcal{R}$-manager. The main contributions of our work are summarized as follows.

- We design a reputation model based on game equilibrium. First, our model uses the confirmed calculation basis to update reputation. Second, the model deducts the vehicles' reputation deposits and taxes them to suppress malicious and selfish behaviors, respectively. Third, the interaction between a service provider and vehicles is modeled as a two-stage single-leader multi-follower Stackelberg game to maximize the utilities of vehicles and service providers.
- We construct a reputation management system based on a consortium blockchain. First, multi-party reputation management activities are abstracted into three types of transactions. Second, we design smart contracts for transactions and achieve multi-party verification of transactions by executing these contracts. Third, we design a half-committee endorsement strategy to further improve system performance.
- $\mathcal{R}$-manager is verified through simulation experiments and system implementation. We prove our reputation model can better suppress malicious and selfish behaviors compared with three different reputation models. Moreover, we implement our reputation management system prototype and it achieves higher throughput and less latency compared with the two blockchain-based reputation management schemes.

The rest paper is outlined as follows. The related work is summarized in Section II. The overview of $\mathcal{R}$-manager is described in Section III. We present our reputation model in Section IV and the reputation management system in Section V. An overall analysis is given in Section VI to demonstrate how $\mathcal{R}$-manager achieves the design goals. In Section VII, $\mathcal{R}$-manager is verified by software simulations and system experiments. Finally, our work is concluded in Section VIII.

II. Related work

In this section, we discuss related work in reputation model design and reputation management system construction. In Table I, we summarize the differences between $\mathcal{R}$-manager and existing works.

A. Reputation Model

The calculation basis is vital to affect the accuracy of vehicle reputation values. Most reputation models rely on the subjective judgment of the task initiator to assess vehicles' reputation values. The authors in [13] calculated the satisfaction ratio from the task initiator based on task completion time and task quality. Then they evaluated the vehicles' reputation based on the satisfaction ratio. A probabilistic model was designed in [11] to infer the quality of the received data and thus obtain the subjective task quality. In addition to subjective judgment, the interaction history of neighboring vehicles is also an important basis for calculating reputation values. The studies in [12] and [21] relied on feedback from other vehicles to calculate reputation values.

To attract vehicles to participate in crowdsourcing, some researchers designed a reputation incentive mechanism [22] to analyze the utilities of vehicles with different behaviors. In order to maximize the utility of the participants, the authors in [10] used the deep learning method to determine the action that can maximize the participants' utilities. The authors in [9] derived game equilibrium after analyzing static and dynamic games in mobile crowdsourcing. The equilibrium-based strategy can attract rational vehicles to actively work. The authors in [23] obtained the optimal pricing under the linear strategic equilibrium in order to maximize the utilities of the participants, thereby providing vehicles with sufficient incentives. In a word, the incentive mechanism can make honest vehicles profitable and attract them to participate in crowdsourcing.

Maximizing the utilities of the vehicles is an important premise for ensuring that vehicles are willing to participate in crowdsourcing. Most reputation management schemes ignore maximizing vehicles' utilities. The authors in [14] and [15] reduced the reputation values of malicious vehicles to force them to report reliably. Still, these methods cannot motivate honest vehicles to participate in crowdsourcing. The authors in [24] utilized Q-learning to help each vehicle make dynamic and optimal decisions in an unknown environment, and vehicles need to report correctly to maximize their rewards.

After analyzing the relevant models, we believe that the following three points should be considered simultaneously when designing a reputation model. (1) The reputation model should use the confirmed calculation basis. Since attackers can provide false judgment or feedback to degrade a vehicle's

TABLE I
COMPARSION WITH OTHER WORKS

No	Reference	Reputation Model				Reputation management system		
		Confirmed calculation basis	Malicious behaviors suppression	Selfish behaviors suppression	Maximizing utilities of SPs and vehicles	Multi-party management	Reputation verification	High system performance
1	[9]	✗	✓	✗	✓	✗	✗	✗
2	[10]	✗	✗	✗	✓	✗	✗	✗
3	[11]	✗	✓	✗	✓	✗	✗	✗
4	[12]	✗	✗	✗	✗	✗	✓	✓
5	[13]	✗	✓	✗	✗	✓	✗	✗
6	[14]	✗	✓	✗	✗	✓	✓	✓
7	[15]	✗	✓	✗	✗	✗	✗	✗
8	[17]	✗	✓	✗	✗	✗	✗	✗
9	[18]	✗	✓	✗	✗	✗	✗	✗
10	[20]	✗	✗	✗	✗	✓	✓	✓
11	[21]	✗	✓	✗	✗	✗	✓	✗
12	[22]	✗	✗	✗	✓	✗	✓	✗
13	[24]	✗	✓	✓	✓	✗	✗	✗
14	[25]	✓	✓	✗	✓	✓	✓	✗
15	[26]	✗	✓	✗	✗	✗	✗	✗
16	[27]	✗	✓	✗	✗	✗	✗	✗
17	[28]	✗	✓	✗	✗	✗	✓	✗
18	[29]	✗	✗	✗	✗	✓	✓	✗
19	[30]	✗	✗	✗	✗	✗	✓	✓
20	[31]	✓	✓	✓	✗	✓	✓	✓
21	$\mathcal{R}$-manager	✓	✓	✓	✓	✓	✓	✓

reputation, a confirmed result rather than a subjective evaluation is required when calculating reputation. (2) The reputation model should resist malicious and selfish behaviors. Since the incentive mechanism is non-coercive, it cannot attract selfish vehicles that refuse to perform tasks. Malicious and selfish behaviors need to be resisted by compulsory measures to make them unable to survive in the network. (3) The reputation model must maximize the utilities of honest vehicles and service providers so that they are willing to participate in crowdsourcing.

B. Reputation Management System

System design is also important to the implementation of vehicle reputation management solutions. The previous methods [15] [26] established a centralized server in the IoV scenario to manage queries and updates of reputation values. Although the idea of building a central server is easy to realize, it has the problems of a single point of failure and poor scalability. With the development of cloud technology, the authors in [27] used a layered reputation management system to calculate the vehicle reputation values on the top layer with the help of fog computing technology. Although the introduction of cloud technology improves computing efficiency and reduces the computing burden of managers, it is still a centralized approach and cannot meet multi-party management needs.

In order to meet the needs of multiple organizations participating in vehicle reputation management and system scalability, some researchers designed a decentralized reputation management system. In [17], each vehicle stored a reputation matrix locally and updated the value of the reputation matrix to maintain the reputation of all vehicles in the network. The authors in [18] used a similar scheme, the difference is that it upgraded the way of storing reputation values to a lightweight database. Although the above-mentioned decentralized reputation management systems constructed the distributed framework, they brought huge overhead to the vehicle. Furthermore, the above methods neither confirm the calculation basis nor reach a consensus on the calculation process. Therefore, it is difficult to uniformly maintain reputation values among distributed nodes, which leads to the inability to provide publicly verifiable reputation services.

To meet the public verification requirements for reputation in traffic-oriented crowdsourcing scenarios, blockchain technology has been introduced to build a reputation management system in recent years. The authors in [29] proposed a blockchain-based hierarchical task management method and stored the vehicle's reputation values on the blockchain. This approach required vehicles to keep a ledger, which puts a strain on the vehicles' limited storage resources. Adaptive fog-blockchain reputation storage was proposed in [28], where fog nodes maintained a blockchain to record the reputation values of all users. In [30], the authors proposed a blockchain-based decentralized framework named CrowdBC, where crowdsourcing activities can be covered by smart contracts. Reputation is updated when executing smart contracts. However, there are two shortcomings in the above methods. First, the existing blockchain-based schemes are not designed for multi-party management needs. In the actual scenario, many parties want to participate in the reputation maintenance process, and the reputation value is only considered trusted by being verified by multiple parties. Second, the consensus protocol based on the public chain consumes a large number of computing resources to submit a transaction, which leads to the inefficiency of the reputation management system.

C. Motivation of $\mathcal{R}$-manager

Aiming at the limitations of previous methods, we make improvements in two aspects by proposing $\mathcal{R}$-manager.

In model design, we have made three innovations. Firstly, only the confirmed calculation basis is used to calculate reputation, which ensures the reputation assessment is convincing. Secondly, our model suppresses malicious and selfish

behaviors with the help of reputation deposit forfeit and a tax mechanism. Thirdly, we model the interaction between a service provider and vehicles as a Stackelberg game. Besides, our model can maximize the utilities of service providers and vehicles by reaching game equilibrium.

In system construction, we have designed a consortium blockchain-based reputation management system for traffic-oriented crowdsourcing in 5G-IoV. First, we design a consortium blockchain to meet the needs of multi-party reputation management. Second, reputation management activities in crowdsourcing applications are abstracted into three types of transactions. Transactions are recorded on the chain to ensure reputation verification and traceability by executing smart contracts. Finally, we propose a half-committee endorsement strategy to improve the system's performance.

It is worth mentioning that we had designed $\mathcal{R}$-tracing [31] to suppress the malicious attacks and selfish behaviors that vehicles may exhibit when reporting messages. However, $\mathcal{R}$-tracing is designed for intelligent traffic applications based on vehicle active reporting, and it is not suitable for traffic-oriented crowdsourcing applications. In a crowdsourcing process, a service provider issues a task, and then the vehicle undertakes and performs the task. Therefore, in addition to vehicles, maximizing the utility of service providers also needs to be considered when building a reputation model. $\mathcal{R}$-tracing cannot meet the need, so we build a reputation model in $\mathcal{R}$-Manager to meet the key need by reaching the Stackelberg game equilibrium.

III. Overview of $\mathcal{R}$-manager

In view of the above shortcomings, we propose $\mathcal{R}$-manager, a consortium blockchain-based vehicle reputation management scheme. In this section, we give an overview of $\mathcal{R}$-manager by introducing the system components, reputation management activities, and our design goals.

A. System Components

As shown in Fig. 1, the system consists of Mobile Operators (MO), the Department of Motor Vehicles (DMV), and the Police Department (PD). We describe each component using the concept of organization. In $\mathcal{R}$-manager, different organizations set up nodes to jointly maintain a consortium blockchain. All nodes within an organization trust each other, but the nodes between organizations do not fully trust other nodes.

MO is responsible for deploying base stations to expand the coverage of 5G signals, thereby expanding crowdsourcing applications to entire urban areas. Macro Base Station (MBS) and Small-cell Base Station (SBS) are two types of base stations deployed by MOs. MBS has better storage and computing capabilities and can be used as a node to maintain a blockchain. The SBS is responsible for forwarding vehicle messages to its MBSes. This configuration ensures that vehicles with poor signals can still communicate with MBS. Therefore, MBSs equipped with multiple SBSes can make up for the limited coverage of 5G signals, which ensures that vehicles can get tasks and return results at any time. In the rest of this paper, we use the base station (BS) to represent an MBS and its affiliated SBSes. In addition to the nodes used for the base station to operate the blockchain, the MO also needs to deploy nodes that provide public services. As shown in Fig. 1, a web service runs on Node 4# will help different service providers issue crowdsourcing tasks. Node 3# and Node 5# act as proxy nodes for two different MOs to participate in reputation management.

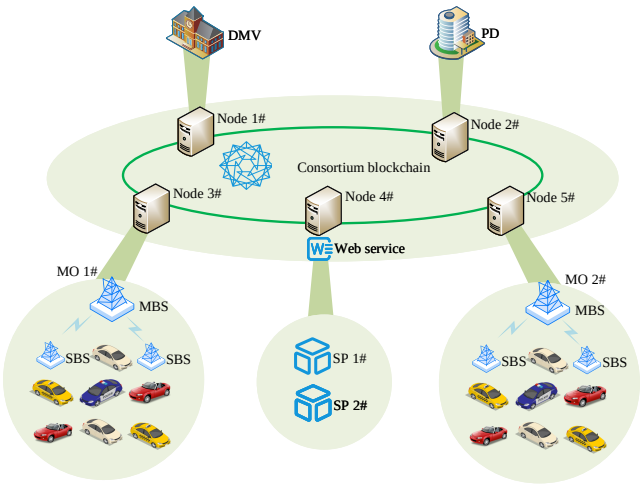


Fig. 1. Consortium blockchain-based reputation management in traffic-oriented crowdsourcing

DMV is responsible for the registration of vehicles entering the network and the cancellation of vehicles leaving the network. A vehicle should apply to the DMV before joining the intelligent transportation application. After receiving a request from a vehicle, the DMV needs to establish a reputation account on the consortium blockchain, assign an initial value, and issue a digital certificate for it. Only vehicles registered with the DMV are legal vehicles. Similar to PD, DMV also sets up some nodes to participate in the maintenance of the blockchain, which will perform reputation calculation result verification.

PD is responsible for the impartial confirmation of a traffic-oriented task result and uploading the confirmed results to the blockchain. The smart contract will be triggered to update the vehicles' reputation values according to the task results confirmed by PD. On the one hand, PD can confirm on the spot by dispatching the police vehicle, such as dealing with vehicle collisions. On the other hand, it can use the existing road traffic monitoring system for direct confirmation, such as observing congestion on a road segment. The task results provided by PD will be uploaded to the blockchain as a truthful calculation basis for the service provider to decide whether to offer rewards. As a party involved in management, PD can verify the transaction of each update of vehicle reputation.

It should be noted that service providers do not participate in the maintenance of the blockchain, and they release task information to nodes that provide public access services to recruit vehicles to complete tasks. Vehicles can also accumulate reputation value through this service. In Fig. 1, two different SPs directly access the web service running on Node 4# to release tasks. Besides, Vehicles as workers for completing

tasks, can be divided into three categories according to their behaviors: honest vehicles, selfish vehicles, and malicious vehicles.

B. Reputation Management Activities

In $\mathcal{R}$-manager, MO, DMV, and PD maintain an official reputation account and manage the reputation values of all vehicles in the network. The service provider can use the crowdsourcing service provided by the blockchain to purchase the corresponding reputation values to reward vehicles. We summarize the reputation management activities for traffic-oriented crowdsourcing into two parts: reputation management based on crowdsourcing tasks and periodic reputation management for vehicles.

The reputation management based on crowdsourcing tasks will update the vehicle's reputation according to the result of the vehicle participating in the crowdsourcing tasks. As shown in Fig. 1, Node 4# is run by a MO and provides public access services. The service provider releases the task to the web service running on Node 4#. Node 4# will find the nearby nodes according to the task location, and forward the task information to them and the corresponding base stations. The base station will broadcast the task information to the vehicles in its area, and vehicles will selectively complete the task according to their situation. Once the task is completed, the vehicle sends the result to the corresponding web service in Node 4# through the base station, and Node 4# will be responsible for uploading task participation records to the blockchain. At the same time, the PD will confirm the authenticity of the task result and update the vehicles' reputation values based on the confirmed result. If the confirmed result is consistent with the result provided by the vehicle, the vehicle's reputation will increase as a reward. Otherwise, the number of false reports will increase by one.

Periodic reputation management for vehicles is used to inhibit the survival of selfish vehicles. Selfish vehicles never participate in crowdsourcing tasks, so no incentive or reward can suppress the vehicles' selfish behaviors. In $\mathcal{R}$-manager, the system periodically collects taxes from vehicles based on their task participation records. After each reputation management period ends, the base station will query the vehicle's historical behaviors and current reputation, calculate the tax of every vehicle, and update the vehicle reputation values.

C. Design Goals of $\mathcal{R}$-manager

To accomplish vehicle reputation management in traffic-oriented crowdsourcing, we propose six design goals that need to be met as follows. The first three of them are for the reputation model and the last three are for the reputation management system.

1) Malicious behaviors suppression: Malicious vehicles can compete with honest vehicles to interfere with normal crowdsourcing. After obtaining task information, a malicious vehicle can choose not to complete the task but upload an arbitrary result to cheat for the reward. Therefore, the reputation model must take into account the malicious behaviors and take appropriate measures to suppress it.

2) Selfish behaviors suppression: Vehicles that always are selfish and unwilling to participate in crowdsourcing should not exist in the system for a long time. They enjoy the convenience brought by the intelligent transportation system without any pay. Therefore, it is necessary to take coercive measures to suppress their survival.

3) Maximizing utilities of SPs and vehicles: In order to attract vehicles to actively participate in crowdsourcing and report high-quality results, the reputation model must ensure that the utilities of honest vehicles are maximized every time they report. Besides, as the basic goal of crowdsourcing, maximizing the utilities of service providers should also be considered.

4) Multi-party management: As shown in **Subsection III-A**, there are different organizations in the vehicle crowdsourcing network, which are trusted within themselves, but not between organizations. Therefore, the need for different organizations to manage reputation values should be considered when designing the reputation management system. Multi-party reputation management cannot only improve the credibility of reputation values but also meet practical needs.

5) Reputation verification: After a reputation value is calculated, it needs to be verified by different organizations. An updated reputation record should be recalculated by each organization to verify that it is the same as the recalculated reputation value. Only if the reputation update record is validated by multiple organizations, then such an updated reputation record can be considered a valid reputation update.

6) High system performance: Most of the previous reputation management systems based on blockchain use public chains, which need to consume plenty of computing resources and inevitably lead to low efficiency of reputation management. To meet the actual reputation management needs, we need to improve system performance and design a reputation management system with high throughput and low latency.

IV. Reputation Model

In this section, we first model the crowdsourcing participation process with the Stackelberg game and then use backward induction to obtain the maximum utility of service providers and vehicles. Finally, We present the additional tax mechanism.

A. Problem Formulation

In the problem definition, we simplify the process of the service provider interacting with the vehicles through the web application. A typical scenario is considered, where a service provider releases some tasks to n vehicles in a region, denoted as $\mathcal{N} = \{1, ..., n\}$. Traffic-oriented crowdsourcing tasks involve identifying congested road sections and confirming collisions at intersections. The vehicle can analyze data from its sensors to give accurate results. The service provider determines the unit price a for every vehicle according to its reputation value. After receiving the task and its price, the vehicles decide whether to engage in this task according to its situation by sending a signal value of e. For all vehicles, the value of e must be greater than 0 and the cost determined

by e must be less than their reputation values. A vehicle pays the corresponding reputation as a deposit if it chooses to participate in some task and reports the task result. When a vehicle is far from the incident of the task, it may decide not to accept this task because of high costs. No reputation will be deducted if a vehicle chooses not to accept tasks. All in all, the whole interaction can be seen as a service provider posting a price for the vehicle and the vehicle making its own decision based on the price. Therefore, the above process is modeled as a two-stage single-leader multi-follower Stackelberg game, where the service provider is the leader and vehicles are followers. The service provider and vehicles all strive to maximize their utilities by choosing the optimal strategy, which ensures that vehicles complete crowdsourcing tasks and report high-quality task results to service providers. The whole game process can be listed in two steps as follows:

Step1: The service provider releases a traffic-oriented task and unit prices $\mathcal{A}$ to these vehicles. $\mathcal{A} = \{a_1, ..., a_n\}$, where a_i is a unit price to be paid for the vehicle i if it participates the task.

Step2: After getting the task information and unit price, every vehicle will choose whether to accept this task. If a vehicle decides to accept this task, it will finish it and send the task result with its signal value e to the service provider. From a vehicle's perspective, it will determine e by maximizing its utility. From a service provider's perspective, the received e marks the vehicle's judgment on the authenticity of the task result, and a larger signal value means the vehicle is more certain of the authenticity of the task result. The vehicle can choose not to accept tasks and keep silent. The service provider receives multiple signal values from vehicles accepting tasks, denoted as $\mathcal{E} = \{e_1, ..., e_n\}$.

Vehicles need to be deducted the reputation value as a deposit according to unit prices and signal values. The deposit paid by the vehicles involved in the task is defined in Eq (1). A vehicle needs to choose an optimal signal value e^* that maximizes its utility.

$$c(e) = ea \tag{1}$$

For any traffic-oriented task, the PD will confirm it and provide a truthful result about this task. The result will be uploaded to the blockchain and the PD will upload rewards according to the vehicle's task completion conditions, and rewards will be paid by the service provider. Rewards will be offered by increasing vehicle reputation if a vehicle completes the task correctly. The reward consists of two parts: the first part is a fixed reward defined in Eq (2), and the second part in Eq (3) is a bonus determined by the signal value submitted by the vehicle. The vehicle will not receive a reward if it reports a wrong task result.

$$r_{fixed} = \alpha log_2 S \tag{2}$$
$$r_{bonus} = r_{max}(1 - e^{-\frac{e}{\beta}}) \tag{3}$$

α and β are two tuning parameters for different applications. Another parameter S in Eq (2) is fixed, which represents the number of vehicle categories in the crowdsourcing network. We also consider selfish vehicles in addition to the honest vehicles and malicious vehicles mentioned in **Subsection III-B**. Therefore, S is set to 3. r_{max} represents the maximum value of the bonus. r_{bonus} enlarges with the increase in e, and its marginal utility [32] will decrease. From the perspective of vehicles, the utility of vehicle v_i with reputation value θ_i is defined in Eq (4). s_i represents whether the vehicle can get the reward, $s_i = 0$ when the result reported by the vehicle is not consistent with the result provided by PD, otherwise $s_i = 1$. A reward is presented by r. The service provider's revenue is equal to the deposit of vehicles minus the reward it provides, and its utility is defined in Eq (5). It should be noted that $\mathcal{R}$-manager also includes a tax mechanism, which is only related to the vehicle's reputation value and irrelevant to the a and e in the utility function. Therefore, we set the tax to be paid by the vehicle as t_i and add it to the utility functions. g indicates whether to trigger the tax mechanism.

$$\begin{aligned} u_i &= s_i r - c_i(e_i) - gt_i \\ &= (\alpha log_2 3 + r_{max}(1 - e^{-\frac{e_i}{\beta}}))s_i - a_i e_i - gt_i \end{aligned} \tag{4}$$

$$\begin{aligned} u_s &= \sum_{i=1}^{n}(a_i e_i - r_i s_i + gt_i) \\ &= \sum_{i=1}^{n} a_i e_i + gt_i - \sum_{i=1}^{n}(\alpha log_2 3 + r_{max}(1 - e^{-\frac{e_i}{\beta}}))s_i \end{aligned} \tag{5}$$

Based on the above analysis, we take **Step 2** of the game as the follower game and transform the vehicle utility maximization problem into the following optimization problem:

$$\begin{aligned} \text{P1}: \quad & \max_{e_i} u_i \\ \textit{s.t.} \quad & C_1 : c_i(e_i) \le \theta_i \\ & C_2 : e_i \ge 0 \end{aligned} \tag{6}$$

For vehicle v_i, two constraints need to be satisfied in Eq (6). First, the vehicle reputation value needs to be guaranteed non-negative after deducting the cost of sending the task result. Second, the signal value selected by the vehicle needs to be greater than or equal to 0.

Step 1 of the game as leader game will be proceeded by the service provider, which is defined in Eq (7). Three constraints need to be considered in the leader game. Like the follower game, the leader game requires two constraints in the follower game for each vehicle. Besides, the third constraint defines that the price needs to be determined by the reputation value of the vehicle, where b_u and b_d respectively represent the upper and lower boundaries.

$$\begin{aligned} \text{P2}: \quad & \max_{e_i, a_i} u_s \\ \textit{s.t.} \quad & C_1 : c_1(e_1) \le \theta_1 \\ & C_2 : e_1 \ge 0 \\ & C_3 : \tfrac{\theta_1}{b_u} \ge a_1 \ge \tfrac{\theta_1}{b_d} \\ & \quad ... \\ & C_{3n-2} : c_n(e_n) \le \theta_n \\ & C_{3n-1} : e_n \ge 0 \\ & C_{3n} : \tfrac{\theta_n}{b_u} \ge a_n \ge \tfrac{\theta_n}{b_d} \end{aligned} \tag{7}$$

The solution of the game represents the optimal strategy of the players. Stackelberg equilibrium will be defined as the game result, which means a player will not obtain more utility if it deviates from the best strategy provided by Stackelberg equilibrium.

Definition 1 (Stackelberg equilibrium): An optimal strategy (a^*, e^*) constitutes a Stackelberg equilibrium if the following conditions are satisfied:

$$u_i(a^*, e^*) \geq u_i(a^*, e_i, e^*_{-i}), \forall e_i \tag{8}$$

$$u_s(a^*, e^*) \geq u_s(a, e^*), \forall a \tag{9}$$

In the above equations, a and e are feasible strategies in the strategy set, and e^*_{-i} is the optimal signal value for all the followers except vehicle v_i. In this way, we model the interaction process between a service provider and the vehicles as a leader-follower game and formalize the utility maximization problem for both. In **Subsection IV-B**, we solve the problem to obtain the optimal actions of the service provider and the vehicle and design the reputation model based on the game equilibrium.

B. Game Equilibrium Solution

Backward induction is a common method to solve the Stackelberg game problem, which is utilized to get the optimal strategy of the problem defined in **Subsection IV-A**. After obtaining the unit price a_i from the service provider, vehicles will compete with each other to maximize their utility. We will prove the existence of a Stackelberg equilibrium in the follower game. According to [33], a Stackelberg equilibrium exists if the following conditions are satisfied.

- The number of players is finite.
- The strategy sets are convex and closed.
- The utility function is continuous and quasi-concave in the set of all feasible solutions.

Our game is a two-stage single-leader multi-follower Stackelberg game. The number of vehicles is n. The strategy set of any vehicle must obey two constraints. The first constraint ensures the signal value e sent by this vehicle is non-negative, which means the vehicle can choose not to undertake the task or choose to complete the task with the confidence e. Therefore, the signal value set is closed and convex. Next, we give **Lemma 1** to prove the existence and uniqueness of optimal strategy.

Lemma 1: Given the unit price $\mathcal{A}$ from the service provider, the optimal strategy for vehicle i can be given as follows:

$$e_i^* = \beta ln \frac{s_i r_{max}}{a_i \beta} \tag{10}$$

Proof: We need to obtain the Hessian matrix of $U_i(a_i, e_i)$ for each follower i. First, we calculate the first derivative of u_i with respect to e_i in Eq (12). Further, the second derivative of u_i with respect to e_i can be defined in Eq (13). Therefore the Hessian matrix of $U_i(a_i, e_i)$ for each follower i can be defined as:

$$H = \begin{bmatrix} \frac{\partial^2 u_i}{\partial e_1^2} & \frac{\partial^2 u_i}{\partial e_1 e_2} & \cdots & \frac{\partial^2 u_i}{\partial e_1 e_N} \\ \frac{\partial^2 u_i}{\partial e_2 e_1} & \frac{\partial^2 u_i}{\partial e_2^2} & \cdots & \frac{\partial^2 u_i}{\partial e_2 e_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 u_i}{\partial e_N e_1} & \frac{\partial^2 u_i}{\partial e_N e_2} & \cdots & \frac{\partial^2 u_i}{\partial e_N^2} \end{bmatrix} \tag{11}$$

The non-diagonal elements of the Hessian matrix are 0, and the diagonal elements of the Hessian matrix are defined in Eq (13). Thus, the matrix is negative semi-definite, and the objective function is concave. So we can conclude there is a unique Stackelberg equilibrium according to the theory in [34]. The proof ends. The best strategy for each follower i can be obtained by setting the first derivative of u_i with respect to e_i equal to zero. The optimal strategy for vehicle i can be computed in Eq (10).

$$\frac{\partial u_i}{\partial e_i} = \frac{s_i r_{max}}{\beta} e^{-\frac{e_i}{\beta}} - a_i \tag{12}$$

$$\frac{\partial^2 u_i}{\partial e_i^2} = -\frac{s_i r_{max}}{\beta^2} e^{-\frac{e_i}{\beta}} < 0 \tag{13}$$

According to the above analysis, the service provider, as a leader in the Stackelberg game, knows the existence of Stackelberg equilibrium among the vehicles. Therefore, Eq (10) can be constituted into Eq (7) and we get a new optimization problem. Obviously, solving this problem is complex because of the large number of constraints. We choose the Gradient Descent algorithm to solve it for Stackelberg equilibrium, which is a common algorithm in optimization problems.

The problem can be solved after using the gradient descent algorithm, which will be demonstrated further in **Subsection VII-A**. As the number of iterations increases, the unit price tends to stabilize. Eventually, we can get the optimal price for the service provider.

$$\begin{aligned} \text{P2}: \quad & \max_{a_i} \textstyle\sum_{i=1}^{n} a_i \beta ln \frac{s_i r_{max}}{a_i \beta} - \sum_{i=1}^{n} (\alpha log_2 3 \\ & + g t_i + r_{max}(1 - e^{-\frac{\beta ln \frac{s_i r_{max}}{a_i \beta}}{\beta}})) s_i \\ \textit{s.t.} \quad & C_1 : a_1 \beta ln \frac{s_1 r_{max}}{a_1 \beta} \leq \theta_1 \\ & C_2 : \beta ln \frac{s_1 r_{max}}{a_1 \beta} \geq 0 \\ & C_3 : \frac{\theta_1}{b_u} \geq a_1 \geq \frac{\theta_1}{b_d} \\ & \quad \cdots \\ & C_{3n-2} : a_n \beta ln \frac{s_n r_{max}}{a_n \beta} \leq \theta_n \\ & C_{3n-1} : \beta ln \frac{s_n r_{max}}{a_n \beta} \geq 0 \\ & C_{3n} : \frac{\theta_n}{b_u} \geq a_n \geq \frac{\theta_n}{b_d} \end{aligned} \tag{14}$$

Based on the game equilibrium, a reputation model will be designed, which specifies the deposit and reward of vehicle participation in the task according to the above equations. Our model relies on game equilibrium and feedback from authority organizations such as PD to ensure that the utilities of vehicles and service providers are maximized. Besides, all the reputation updates will be verified by nodes from different organizations in a blockchain network, which ensures the authenticity of the calculated reputation value.

C. Tax Mechanism

As an additional module of our model, the tax mechanism is used to discourage selfish behaviors. It needs to comply with two fundamental principles. The first principle is that vehicles with a high reputation value should pay a higher tax than vehicles with a low reputation value. $\mathcal{R}$-manager considers situations where vehicles, that have obtained high reputation values, stop receiving tasks. The first principle is to prevent the situation described above. It ensures that the vehicle's reputation value cannot remain stable all the time. When reputation decays, selfish vehicles are compelled to participate

in crowdsourcing tasks. This principle also corresponds to the real world: high-income people should pay more taxes and take more social responsibilities. The second principle is that if a vehicle reports more wrong crowdsourcing results, then the tax it should pay should be higher. Records of vehicles participating in crowdsourcing tasks are stored on the blockchain, which means that the vehicle activities have the consensus of multiple organizations. Taking vehicle i as an example, the consortium blockchain records the total number of times p_{ia} that vehicle i participates in crowdsourcing tasks and the number of times p_{ir} that vehicle i participates honestly in crowdsourcing tasks. The number of times p_{iw} that vehicle i participates maliciously in tasks can be computed with p_{ia} and p_{ir}. So we can define tax mechanism as follows.

$$t_i(\theta_i, \delta_i, d_i) = \begin{cases} r_1 \frac{\theta_i}{\sum_{i=1}^{n_1} \theta_i} T & \frac{p_{ir}}{p_{ia}} >= thr_1 \\ r_2 \frac{p_{iw}}{\sum_{i=1}^{n_2} p_{iw}} T & \frac{p_{ir}}{p_{ia}} < thr_1 \end{cases} \tag{15}$$

In the above equation, thr_1 represents the confidence threshold. At the beginning of each tax cycle, the base station will classify all vehicles into two vehicle sets according to their participation in the crowdsourcing task. The first set is the potentially honest vehicle set (PHVS). Vehicles i will be considered potentially honest if $\frac{p_{ir}}{p_{ia}}$ is greater than thr_1. Otherwise, it will be classified as a potentially malicious vehicle set (PMVS). For a vehicle in PHVS, the tax will be charged according to its reputation value, and a vehicle with a higher reputation value will pay more tax. In PMVS, the tax is computed according to the number of maliciously submitted crowdsourcing results. Vehicles that maliciously participate in crowdsourcing tasks more times will be charged more reputation values as tax. n_1 and n_2 represent the number of vehicles in the above two vehicle sets respectively. Besides, T represents the tax to be levied by service providers since the last tax period. r_1 and r_2 need to satisfy the following equations.

$$r_1 + r_2 = 1 \tag{16}$$

V. Reputation Management System

This section presents the detail of the reputation management system based on a consortium blockchain. Three different types of transactions are abstracted to fulfill the needs of reputation management activities. Besides, all the transactions will be as the basic data structure operated by smart contracts, which will automatically complete reputation calculation and verification.

A. Transaction Design

In order to accomplish reputation updates and reputation verification, we design three types of transaction structures as shown in Fig. 2. Type 1 transactions are used to extract evidence from the results of crowdsourcing tasks reported by the vehicle, and Type 2 transactions are used to reward vehicles with correct task results. Type 3 transactions are initiated by the base station to collect tax from the vehicle. The common part of the three types of transactions is the green area, where *TransId* is the identifier assigned to each transaction, *Type* marks the category of a transaction, and *NodeId* marks the transaction initiator. *VehicleId* represents the vehicle identifier. Besides, *UploadTimestamp* records the time the transaction was uploaded.

In the Type 1 transaction, *SignalValue* records the signal value a vehicle chooses. *Cost* is the guaranteed reputation value to be paid for completing this task. *TaskId* specifies the task identifier, *Position* records the task position and *SndTimestamp* records the timestamp of the transaction sent. In the Type 2 transaction, *Result* marks whether the task result provided by the vehicle is consistent with the result after PD confirmation. *UpdateValue* records the specific reputation value of this operation. *TaskId* specifies the task identifier corresponding to this transaction. In the Type 3 transaction, *Right* and *Sum* represent the number of times the vehicle has correctly participated in crowdsourcing tasks and the number of times the vehicle has historically participated in crowdsourcing tasks, respectively. The tax to be paid on the vehicle is saved in the *Tax*. The last *PeriodEndTime* records the last tax timestamp.

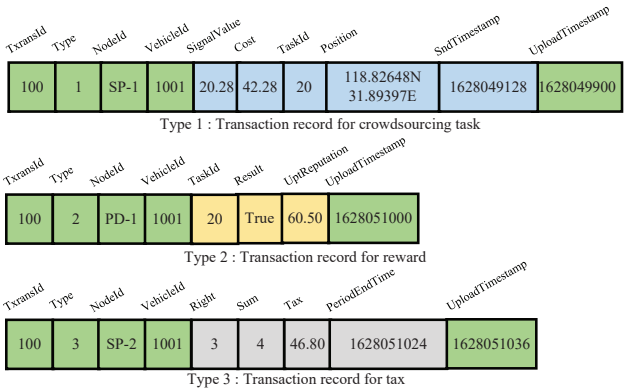


Fig. 2. Transaction records for reputation management

B. Half Committee Endorsement Strategy

The reputation management system based on PoW [19] requires nodes to spend a lot of computing resources to reach a consensus. This process is time-consuming, so it cannot meet the reputation management needs of actual crowdsourcing scenarios. $\mathcal{R}$-manager contains a consortium blockchain-based reputation management system, replacing the consensus of all nodes with the consensus among organizations, transactions can be uploaded to the chain in a shorter time. In a consortium chain, the endorsement strategy is used to reach a consensus among organizations. To further improve the system throughput while ensuring the correctness of transactions, the half-committee endorsement strategy is proposed. The core idea of the half-committee is to select endorsement nodes based on the number of organizations that participate in the maintenance of the blockchain. These selected nodes will form a committee. A transaction is valid only after the endorsement of a transaction is completed in every committee node. We assume that the attacker cannot know or manipulate the randomly selected organizations or their nodes. Therefore, the committee can be considered trustworthy.

Algorithm 1: Half Committee Endorsement Strategy

Input: *trd*: transaction record; N_{org}: the number of organizations;
Output: *rlt*: endorsement result;

$N_{end} \leftarrow \lfloor \frac{N_{org}}{2} \rfloor + 1$;
cnt $\leftarrow$ 0;
$\mathbb{O} \leftarrow \{\}$;
$\mathbb{E} \leftarrow \{\}$;
for i = *1 To* N_{end} **do**
    $o_i \leftarrow$ ***get_random_node***(i);
    $\mathbb{O}$.***add_node***(o_i);
for o_i *in* $\mathbb{O}$ **do**
    $e_i \leftarrow$ ***get_random_node***(o_i);
    $\mathbb{E}$.***add_node***(e_i);
for e_i *in* $\mathbb{E}$ **do**
    switch (*trd.type*) **do**
        case *1* **do**
            $f \leftarrow e_i$.***call_smart_contract_1***(*trd*);
            break;
        case *2* **do**
            $f \leftarrow e_i$.***call_smart_contract_2***(*trd*);
            break;
        case *3* **do**
            $f \leftarrow e_i$.***call_smart_contract_3***(*trd*);
            break;
    if f **then**
        cnt $\leftarrow$ *cnt* + 1;
if (*cnt* = N_{end}) **then**
    rlt $\leftarrow$ true;
else
    rlt $\leftarrow$ false;
return *rlt*;

Algorithm 2: Smart contract for Type 1 transactions

Input: trd: transaction record;
Output: rlt: execution result;

$t \leftarrow$ ***get_task_from_node***(*trd.nodeId,trd.transId*);
$\theta \leftarrow$ ***get_reputation_from_chain***(*trd.vehicleId*);
if (*trd.signalValue* = *t.signalValue*) **then**
    $c \leftarrow$ ***cal_cost***(*trd.signalValue*,θ);
    if ($c > \theta$ OR $c < 0$) **then**
        rlt $\leftarrow$ false;
    else if ($c = trd.cost$) **then**
        rlt $\leftarrow$ true;
    else
        rlt $\leftarrow$ false;
else
    rlt $\leftarrow$ false;
return *rlt*;

Algorithm 3: Smart contract for Type 2 transactions

Input: trd: transaction record;
Output: rlt: execution result;

rlt $\leftarrow$ true;
$\theta \leftarrow$ ***get_reputation_from_chain***(*trd.vehicleId*);
$e \leftarrow$ ***get_signal_from_chain***(*trd.taskId*);
if (*trd.Flag* = True) **then**
    $w \leftarrow$ ***cal_reward***(e);
    if (w = *trd.updateValue*) **then**
        $\theta' \leftarrow \theta + w$;
    else
        rlt $\leftarrow$ false;
return *rlt*;

As shown in Algorithm 1, the half-committee endorsement strategy is divided into three main steps: (1) The algorithm computes the number of endorsement nodes according to the number of organizations, and randomly selects organizations according to the number of endorsement nodes. (2) Each selected organization randomly selects endorsement nodes and adds them to the endorsement set built for the current transaction. (3) Each endorsement node executes the corresponding smart contract to confirm the transaction. After completing the execution of the endorsement strategy, the endorsement process of the current transaction ends, which ensures that each transaction is confirmed.

C. Smart Contract for Reputation Management

The smart contract is automatically executed in multiple nodes from different organizations, and the process is called an endorsement. Every transaction defined in **Subsection V-A** is valid only after different nodes have executed smart contracts to verify it. The system can be deployed with a pre-defined number of nodes to endorse a transaction. We design three smart contracts for the three types of transactions to manage reputation, and each node will execute the corresponding smart contract to complete an endorsement when it receives a transaction.

Algorithm 2 will be executed when the node receives a Type 1 transaction. The endorsing node first obtains the original task information based on the *TransId* and then gets the vehicle reputation value based on the *VehicleId*. Secondly, the endorsing node checks whether the signal value in the transaction and the signal value in the task information are the same, only if they are the same will it calculate the cost. After calculating the cost, a cost legality check that ensures the calculated deposit cannot exceed the reputation value will be conducted. Finally, the node compares whether the calculated deposit and the cost in the transaction *trd* are the same, if they are then it returns true. The algorithm will output false if the transaction does not pass any of the above checks.

According to steps in Algorithm 3, the endorsing node first gets the current reputation value of the specified vehicle and queries the signal value from the chain. Then it calculates the reward for the vehicle according to the confirmation result. If the calculated reputation value is consistent with the *UpdateValue*, it will update the reputation value and return true. Otherwise, it returns false.

Algorithm 4 is called periodically to get the tax from the vehicle. Firstly, the endorsing node obtains the current reputation of the vehicle, the number of tasks completed, and the number of tasks completed correctly. Secondly, con is calculated to decide which category the vehicle belongs to. Finally, the tax to be paid on the vehicle will be calculated and compared with the tax claimed in the transaction, returning true if it is the same and false otherwise.

Multiple organization nodes complete the transaction verification by running the smart contract, and the transaction includes the update process of reputation values. Therefore the entire reputation calculation process is open and transparent, so the reputation value has strong credibility.

Algorithm 4: Smart contract for Type 3 transactions

Input: trd: transaction record;
Output: rlt: execution result;
$\theta \leftarrow$ ***get_reputation_from_chain***(*trd.vehicleId*);
$T \leftarrow$ ***get_official_account_change***(*trd.periodEndTime*);
$T_1 \leftarrow r_1 \cdot T$;
$T_2 \leftarrow r_2 \cdot T$;
$p_r \leftarrow$ ***get_right_from_chain***(*trd.vehicleId*);
$p_a \leftarrow$ ***get_all_from_chain***(*trd.vehicleId*);
$con \leftarrow$ ***cal_confidence***(p_r,p_a) ;
if ($con > thr_1$) **then**
  type $\leftarrow$ 1;
else
  type $\leftarrow$ 2;
$t \leftarrow$ ***cal_tax***(*type*,p_r,p_a);
if ($t > \theta$ OR $t < 0$) **then**
  rlt $\leftarrow$ false;
else if t = *trd.tax* **then**
  rlt $\leftarrow$ true;
else
  rlt $\leftarrow$ false;

VI. Analysis of $\mathcal{R}$-manager

In this section, we demonstrate $\mathcal{R}$-manager can fulfill all the design goals listed in **Subsection III-C**.

1) Malicious behaviors suppression: We model the interaction between vehicles and a service provider as a two-stage single-leader multi-follower game, where both the service provider and vehicles will take actions that maximize their utility. The utility of a malicious vehicle under different actions is analyzed. u_m^t and u_m^f represent the revenue of a malicious vehicle reporting a correct task result and reporting a wrong task result.

$$\begin{aligned} u^t &= r - c_i(e_i) - gt_i \\ &= \alpha log_2 3 + r_{max}(1 - e^{-\frac{e_i}{\beta}}) - a_i e_i - gt_i \end{aligned} \tag{17}$$

$$u^f = -c_i(e_i) - gt_i \tag{18}$$

In Eq (18), the vehicle's utility is the reward he gets minus the cost of sending the task result, and the vehicle cannot get any reward in Eq (19). Obviously, u^t is always bigger than u^f. So the best choice for malicious vehicles is to report correct task results. In our model, the malicious vehicle cannot get any reward and pay the deposit. Besides, the additional tax mechanism will deduct its reputation. Therefore, malicious behaviors suppression is achieved through deducting the deposit and taxing.

2) Selfish behaviors suppression: In **Subsection IV-C**, we design a tax mechanism that divides vehicles into two categories based on crowdsourcing task completion results. vehicles that mainly behave honestly will pay tax according to their own reputation, and vehicles that mainly report maliciously will be charged according to the frequency of malicious behaviors. The purpose of our tax mechanism is to impose reputation deductions on all vehicles to force them actively participate in crowdsourcing tasks. For vehicles that do not participate in the crowdsourcing task, their $\frac{p_{ir}}{p_{ia}}$ will be regarded as 0. They are classified as PMVS and their utility is computed in Eq (20). So our model suppresses selfish behaviors because they cannot maintain their reputation at a high level for a long time.

$$u^s = -t_i(\theta_i, \delta_i, d_i) = \begin{cases} -r_1 \frac{\theta_i}{\sum_{i=1}^{n_1} \theta_i} T & \frac{p_{ir}}{p_{ia}} >= thr_1 \\ -r_2 \frac{p_{iw}}{\sum_{i=1}^{n_2} p_{iw}} T & \frac{p_{ir}}{p_{ia}} < thr_1 \end{cases} \tag{19}$$

3) Maximizing utilities of SPs and vehicles: We model the interaction between vehicles and a service provider as a Stackelberg game and solve their utility maximization problem to obtain the Stackelberg game equilibrium. Because both honest vehicles and service providers perform crowdsourcing tasks based on the game equilibrium, their utilities can be maximized.

4) Multi-party management: In $\mathcal{R}$-manager, we have built a distributed reputation management system for different organizations, in which DMV, PD, and MOs jointly maintain the entire system. Each node from all the organizations forms a consortium chain. The submission of transactions on the chain requires endorsement by different organizations. A transaction submitted by a node is only valid after other nodes reach a consensus. Therefore, $\mathcal{R}$-manager's reputation management system can meet the needs of multi-party management.

5) Reputation verification: In $\mathcal{R}$-manager, reputation verification is completed by executing smart contracts to verify transactions. Due to the immutable nature of the blockchain, transactions can provide a reliable calculation basis for reputation verification. Any node can execute smart contracts to check whether the reputation value in the transaction is correct, so $\mathcal{R}$-manager can provide verifiable reputation service for different organizations.

6) High system performance: To meet the practical requirements of reputation management, it is necessary to improve the system's performance. $\mathcal{R}$-manager has made two optimizations. First, a reputation management system based on the consortium chain is designed, which replaces the consensus of nodes with the consensus among organizations for improving system efficiency. Secondly, the half-committee endorsement strategy is designed to improve system efficiency.

VII. Experiments

In this section, we conduct experiments including two parts to validate the effectiveness of $\mathcal{R}$-manager. The first part is to prove that the Stackelberg equilibrium-based reputation model can effectively discourage malicious and selfish workers and attract honest workers to report high-quality results. The second part is to build a prototype system and test its performance. Besides, we share the source code of $\mathcal{R}$-manager on GitHub to make it beneficial.[1]

A. Experiment Analysis for Reputation Model

The reputation model aims to suppress malicious and selfish behaviors and attract honest vehicles to report high-quality task results. We need to first build a traffic-oriented crowdsourcing environment and implement the designed reputation model. To build a crowdsourcing environment with as many vehicles as possible in IoV, our simulation is conducted on MATLAB

[1]https://github.com/enzeyu/TVT_R-manager

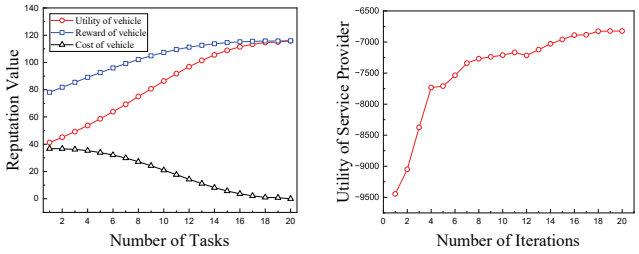
TABLE II
PARAMETERS IN THE EXPERIMENT

Parameter	Value
α	10
β	2
r_{max}	100
b_u	10
b_d	20
Number of vehicles	50,50,100,100,150
Number of tasks	50
Number of service provider	1
Malicious rate	10%,20%,30%,40%,50%
Tax interval	2
The maximal reputation	1000

R2020b, where five scenes with different numbers of vehicles, different malicious rates, and one service provider are constructed as follows:

- Scene 1: The number of vehicles is 50, and the malicious rate is 10% (5 vehicles).
- Scene 2: The number of vehicles is 50, and the malicious rate is 20% (10 vehicles).
- Scene 3: The number of vehicles is 100, and the malicious rate is 30% (30 vehicles)
- Scene 4: The number of vehicles is 100, and the malicious rate is 40% (40 vehicles).
- Scene 5: The number of vehicles is 150, and the malicious rate is 50% (75 vehicles).

Besides, 50 crowdsourcing tasks are released to simulate a long-running crowdsourcing platform in five scenes. The initial reputation value follows a normal distribution with a mean of 400 and a variance of 50. The parameters defined in Section III are shown in Table II. We set the tax interval to 2, which means the tax mechanism will be triggered automatically after every two missions. Experiments demonstrate how different vehicles' reputation values change under different reputation models as crowdsourcing tasks increase. We discuss the average results of 10 repeated experiments in the following paragraphs.



(a) Utility of a vehicle (b) Utility of the service provider

Fig. 3. Reputation model convergence

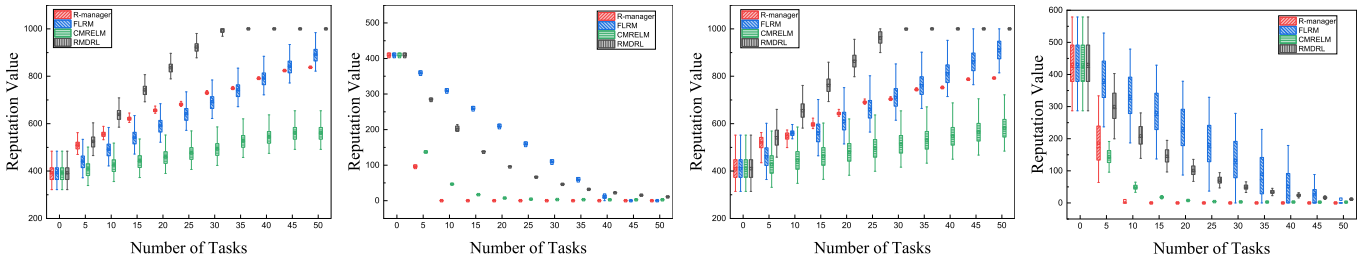
The convergence of the game equilibrium-based reputation model is the first part that needs to be proven. As shown in Fig. 3b, the utility of the service provider gradually increases as the number of iterations increases and stabilizes at the 18# iteration. Because the service provider needs to provide reputation values, the maximization of its utility is equal to the loss minimization. Then we take an honest vehicle as an example and analyze its utility change with the increase in the number of tasks, which proves our model can motivate vehicles to engage in crowdsourcing tasks. As the number of participation tasks increases, this vehicle's participation cost decreases, and its reputation value increases. The reward the vehicle receives increases and then stabilizes, which is shown in Fig. 3a. Therefore, our model can converge to a steady state.

Further, we conduct comparative experiments to show that our model can better suppress malicious behaviors and attract honest vehicles to report high-quality task results. We select three different methods as follows:

- Fixed Linear Reputation Model (FLRM) [30] [35]: The idea of FLRM is that vehicles receive a fixed reward for completing tasks honestly and a penalty of the same value for completing tasks maliciously.
- Computational Model for Reputation and Ensemble-based Learning Model (CMRELM) [36]: A type of nonlinear reputation model that uses piecewise functions to update reputation.
- Reputation Mechanism-based Deep Reinforcement Learning (RMDRL) [37]: The idea of RMDRL is to obtain the optimal vehicle action based on reinforcement learning.

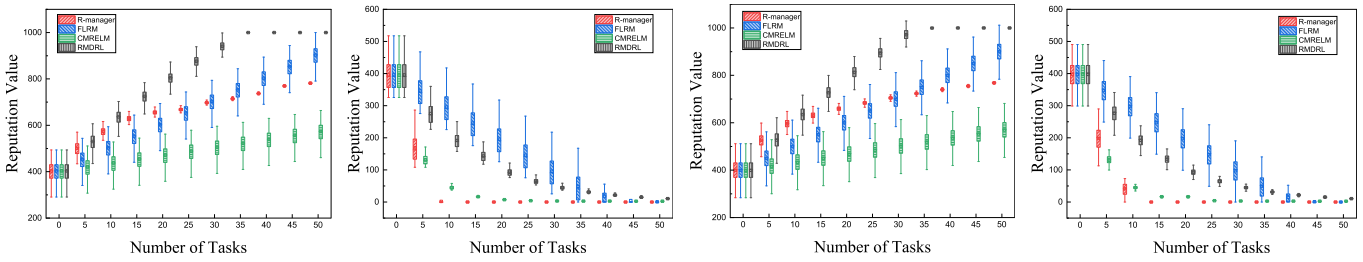
Fig. 4-Fig. 6 show the impact of four reputation models on the reputation values of honest and malicious vehicles in five scenes. We can observe that the effects of the four reputation models on honest vehicles and malicious vehicles are similar in different scenes. We take Scene 3 as an example for analysis. As shown in Fig. 5a, the reputation values of honest vehicles increase as the number of participating tasks increases under the influence of all reputation models. The reason for this phenomenon is that s_i of honest vehicles is 1, which will be rewarded accordingly. However, due to the selection of optimal actions in RMDRL, the vehicle's reputation value reaches maximum value after 35 tasks, which may lead to selfish behavior of the vehicle. FLRM also increases the vehicle's reputation value without any limit. CMRELM does not give vehicles sufficient rewards, making them with insufficient motivation to participate in crowdsourcing. In contrast, $\mathcal{R}$-manager strikes an appropriate balance between rewarding honest vehicles and suppressing selfish behavior. $\mathcal{R}$-manager controls the growth rate of vehicle reputation value through tax mechanisms and reward mechanisms to force vehicles with high reputation values to participate in crowdsourcing. As shown in Fig. 5b, all models can suppress malicious behaviors by reducing vehicle reputation but only $\mathcal{R}$-manager ensures that each vehicle's reputation can decrease at a faster rate than other methods through collecting taxes and deducting reputation deposits. In our model, s_i of malicious vehicles is 0, so they get no reward. As the number of tasks increases, the cost paid by malicious vehicles becomes higher, which eventually leads to their reputation value becoming 0.

To show the superiority of the $\mathcal{R}$-manager more clearly, we choose two example vehicles to demonstrate the reputation change, where vehicle 23# is an honest vehicle and vehicle 8# is a malicious vehicle. Fig. 7a displays the reputation value changes of vehicle 23#. During the first 20 tasks, the growth



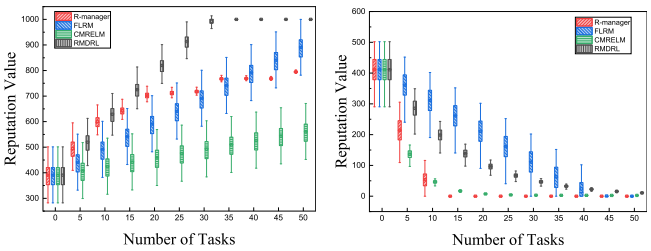
(a) Honest vehicles (Scene 1) (b) Malicious vehicles (Scene 1) (c) Honest vehicles (Scene 2) (d) Malicious vehicles (Scene 2)

Fig. 4. Reputation value changes of honest and malicious vehicles in four models (Scene 1 and Scene 2)



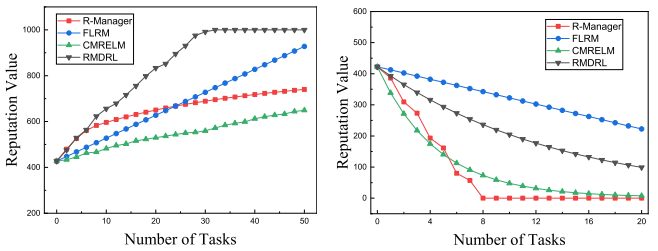
(a) Honest vehicles (Scene 3) (b) Malicious vehicles (Scene 3) (c) Honest vehicles (Scene 4) (d) Malicious vehicles (Scene 4)

Fig. 5. Reputation value changes of honest and malicious vehicles in four models (Scene 3 and Scene 4)



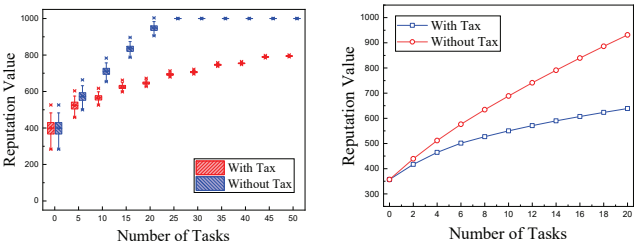
(a) Honest vehicles (Scene 5) (b) Malicious vehicles (Scene 5)

Fig. 6. Reputation value changes of honest and malicious vehicles in four models (Scene 5)



(a) Vehicle 23# (b) Vehicle 8#

Fig. 7. Reputation value changes of vehicle 8# and vehicle 23# in four models



(a) Honest vehicles (b) Vehicle 18#

Fig. 8. Reputation value changes of vehicles under the influence of the tax mechanism

rate of $\mathcal{R}$-manager is higher, and after 20 tasks, the growth of reputation value will be slowed. In contrast, both FLRM and RMDRL reward honest vehicles infinitely, and although the CMRELM method increases rewards in subsequent tasks, it is difficult to attract honest vehicles to participate in early tasks due to the low rewards. Therefore, $\mathcal{R}$-manager achieves a good balance between rewarding honest vehicles and inhibiting selfish vehicles. Fig. 7b shows the reputation value change of vehicle 8#. In $\mathcal{R}$-manager, under the dual effects of the tax mechanism and reputation deposit, the malicious vehicle's reputation value drops rapidly and will decrease to 0 after 8 tasks. FLRM punishes malicious vehicles with a fixed reputation value, so the punishment is insufficient to suppress them. The CMRELM and RMDRL methods cannot exert severe penalties when the malicious vehicles' reputation value is low, which increases the probability of these vehicles launching malicious attacks. In summary, $\mathcal{R}$-manager outperforms other methods in suppressing malicious vehicles and attracting honest vehicles to complete crowdsourcing tasks.

Lastly, we prove our tax mechanism plays a vital role in preventing selfishness. We first show 40 honest vehicles' reputation value changes under tax and no taxes. As shown in Fig. 8a, although the reputation values of honest vehicles are generally growing, their growth rate has slowed down under the effect of the tax mechanism. The reputation values of vehicles remain relatively stable after reputation reaching 800. If there is no tax, the reputation values of vehicles will be close to the maximum reputation after participating in 20 crowdsourcing tasks. The vehicles may show selfish behaviors because their reputation values are relatively high. Fig. 8b

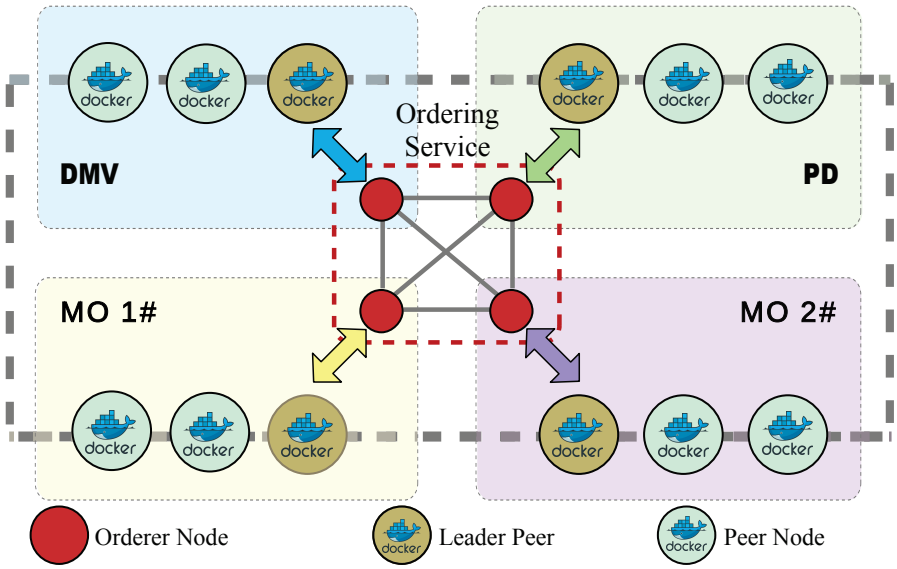


Fig. 9. Reputation management system implementation by consortium blockchain

takes vehicle 18# as an example to display the impact of the tax mechanism on the vehicle's reputation. For vehicles that remain selfish, it is clear that they cannot survive in crowdsourcing due to their regular tax payments. In general, the tax mechanism can further enable vehicles to participate in more crowdsourcing tasks and reduce the occurrence of selfish behaviors.

B. Experiment Analysis for Prototype System

We use Vmware software to build a 4GB RAM virtual machine as an experimental platform, which has 40G external memory and a dual-core processor. Our system is implemented on Ubuntu 16.04 with tools including Docker and Hyperedger Fabric v2.2.0 with 1184 lines of core codes, among which 330 lines are for implementing smart contracts mentioned in **Subsection V-C** with Golang language. A reputation management system implementation is shown in Fig. 9, where four leader nodes belonging to PD, DMV, and two MOs are organized to simulate the scenario where multiple organizations manage reputation. Each organization also sets up two peer nodes for providing external services and one orderer node for sorting transactions. In Hyperledger Fabric, the orderer service is responsible for sorting transactions and packing them into blocks, which is utilized by setting orderer nodes in different organizations. In the actual implementation, due to the limitation of system resources, we only design one orderer node and four leader nodes. The leader node will take responsibility for providing external services. The consensus mechanism used by the reputation management system is the Raft protocol. In $\mathcal{R}$-manager, after the half-committee endorsement strategy verifies the correctness of transactions, the Raft protocol is used to complete the transaction upload.

For the system to meet actual needs, all smart contracts are packaged as web interfaces. Interfaces and their functions are shown in Table III. *record_veh* interface encapsulates Algorithm 2, recording vehicle participation in crowdsourcing.

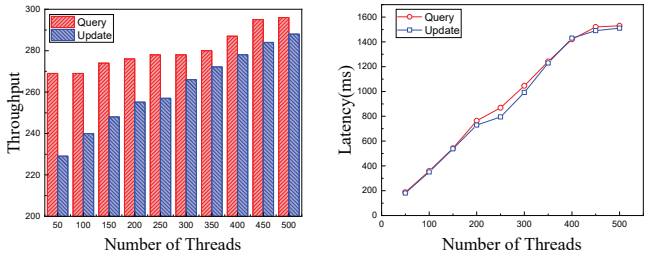
TABLE III
INTERFACES IN REPUTATION MANAGEMENT SYSTEM

Interface name	Function	Encapsulation
register_veh	Register vehicles	
delete_veh	Revoke vehicles	
record_veh	Record task participation	Algorithm 2
update_veh	Update vehicle information	Algorithm 3
tax_veh	Tax vehicles	Algorithm 4
query_veh	Query vehicle information	

update_veh interface encapsulates Algorithm 3, providing the reputation update service. *tax_veh* interface encapsulates Algorithm 4, providing the tax service. It should be noted that the remaining three interfaces do not involve the reputation model, so no specific algorithm is encapsulated. According to interface functions and actual scene needs, querying and updating reputation values are the two most commonly used services. Algorithm 2 records the conditions of vehicles participating in crowdsourcing, and it corresponds to updating a record. Algorithm 3 and Algorithm 4 both use the query interface to get relevant vehicle information and then update the vehicle reputation values. Therefore, in the following experiments, we mainly measure the system's performance in terms of updates and queries.

Throughput and latency are used as performance metrics in the experiment. Throughput is the number of requests processed by the system per unit of time. Latency is the time difference between when a request is sent and when the result is obtained. High throughput means the system can handle more requests and low latency means the request will be returned in a short period. In the experiment, different thread numbers are constructed to simulate the actual scenario of multiple users performing reputation management. Every thread establishes a connection and sends a request. The number of threads is adjusted at intervals of 50 from 50 to

500 to send requests. Experiment results will be analyzed as follows.



(a) Throughput (b) Latency

Fig. 10. Throughput and latency under different thread numbers

As shown in Fig. 10a. The throughput of the query reputation increases with the rise in the number of threads. When the number of threads increases from 450 to 500, the throughput hardly changes, because the entire system reaches a bottleneck state. Reputation value update also has a similar trend. But reputation update throughput is lower than the query throughput. The reason is reputation update requires two operations of reading and writing. When the whole system reaches the bottleneck state, the query throughput and update throughput are 296 and 288, respectively. As shown in Fig. 10b, the system's query latency and update latency increase to more than 1000ms as the number of threads rises. The latency is around 1.5 seconds when the system reaches a bottleneck state, which is acceptable.

In summary, our consortium blockchain-based management system can meet practical management needs by achieving a throughput of 280 and a latency of around 1.5 seconds in a machine. In [20], the proposed scheme adopts the PoW. Its throughput is only 12 since many computing resources are used for solving complex problems, The similar system is constructed [30], which relies on Ethereum technology to achieve a system with 37 seconds delay. The above methods cannot meet the practical reputation management needs in IoV. We design a half-committee endorsement strategy, which improves system throughput and reduces latency by reducing the time to reach consensus.

VIII. Conclusion

Obtaining high-quality reporting in the presence of malicious and selfish vehicles is gaining attention for traffic-oriented crowdsourcing in 5G-IoV. Previous reputation-based works still have some drawbacks. In this paper, we propose a consortium blockchain-based vehicle reputation management scheme named $\mathcal{R}$-manager. In model design, we design a reputation model based on feedback from the authority department to suppress malicious behaviors by reputation deposit forfeit. Besides, a tax mechanism is proposed to curb selfish vehicles. Our model also maximizes the utilities of service providers and vehicles by reaching Stackelberg game equilibrium. In system construction, we propose a multi-organization reputation management architecture and design three transactions to cover all the reputation management activities. Besides, we design three smart contracts to provide reputation verification and management. To improve system performance, we design a half-committee endorsement strategy. Simulation experiments prove our model can effectively attract honest vehicles to report high-quality task results by maximizing vehicles' utility and repressing malicious and selfish behaviors. Besides, a reputation management system is constructed and achieves 280 transaction processes per second and a latency of up to 1.5 seconds.

References

[1] W. Duan, J. Gu, M. Wen, G. Zhang, Y. Ji, and S. Mumtaz, "Emerging technologies for 5g-iov networks: applications, trends and opportunities," *IEEE Network*, vol. 34, no. 5, pp. 283–289, 2020.

[2] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "Lotad: Long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, pp. 825–847, 2018.

[3] W. Guo, Z. Chang, Y. Su, X. Guo, T. Hämäläinen, J. Li, and Y. Li, "Reputation-based blockchain for spatial crowdsourcing in vehicular networks," *Applied Sciences*, vol. 12, no. 21, p. 11049, 2022.

[4] W. Chen, Y. Chen, X. Chen, and Z. Zheng, "Toward secure data sharing for the iov: A quality-driven incentive mechanism with on-chain and off-chain guarantees," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1625–1640, 2019.

[5] W. Feng, Z. Yan, L. T. Yang, and Q. Zheng, "Anonymous authentication on trust in blockchain-based mobile crowdsourcing," *IEEE Internet of Things Journal*, 2020.

[6] J. Zhang, F. Yang, Z. Ma, Z. Wang, X. Liu, and J. Ma, "A decentralized location privacy-preserving spatial crowdsourcing for internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2299–2313, 2020.

[7] H. Mousa, S. B. Mokhtar, O. Hasan, O. Younes, M. Hadhoud, and L. Brunie, "Trust management and reputation systems in mobile participatory sensing applications: A survey," *Computer Networks*, vol. 90, pp. 49–73, 2015, crowdsourcing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128615002340

[8] R. Hussain, J. Lee, and S. Zeadally, "Trust in vanet: A survey of current solutions and future research opportunities," *IEEE transactions on intelligent transportation systems*, vol. 22, no. 5, pp. 2553–2571, 2020.

[9] L. Xiao, T. Chen, C. Xie, H. Dai, and H. V. Poor, "Mobile crowdsensing games in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1535–1545, 2017.

[10] Y. Yue, W. Sun, J. Liu, and Y. Jiang, "Ai-enhanced incentive design for crowdsourcing in internet of vehicles," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–5.

[11] Y. Zhao, X. Gong, and X. Chen, "Privacy-preserving incentive mechanisms for truthful data quality in data crowdsourcing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 7, pp. 2518–2532, 2022.

[12] W. Feng and Z. Yan, "Mcs-chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain," *Future Generation Computer Systems*, vol. 95, pp. 649–666, 2019.

[13] Z. Wang, L. Liu, L. Wang, X. Wen, and W. Jing, "Privacy-protecting reputation management scheme in iov-based mobile crowdsensing," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2020, pp. 337–343.

[14] L. Sun, Q. Yang, X. Chen, and Z. Chen, "Rc-chain: Reputation-based crowdsourcing blockchain for vehicular networks," *Journal of Network and Computer Applications*, vol. 176, 2020.

[15] R. Shrestha, R. Bajracharya, and S. Y. Nam, "Centralized approach for trustworthy message dissemination in vanet," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–5.

[16] H. Hu, R. Lu, Z. Zhang, and J. Shao, "Replace: A reliable trust-based platoon service recommendation scheme in vanet," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1786–1797, 2016.

[17] W. Li and H. Song, "Art: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 960–969, 2015.

[18] F. Ahmad, F. Kurugollu, A. Adnane, R. Hussain, and F. Hussain, "Marine: Man-in-the-middle attack resistant trust model in connected vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3310–3322, 2020.

[19] C. Zhang, Y. Guo, H. Du, and X. Jia, "Pfcrowd: Privacy-preserving and federated crowdsourcing framework by using blockchain," in *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, 2020, pp. 1–10.

[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 Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3616–3630, 2020.

[21] Y. Liu, Z. Xiong, Q. Hu, D. Niyato, J. Zhang, C. Miao, C. Leung, and Z. Tian, "Vrepchain: A decentralized and privacy-preserving reputation system for social internet of vehicles based on blockchain," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 12, pp. 13 242–13 253, 2022.

[22] J. Cheng, H. Long, X. Tang, J. Li, M. Chen, and N. Xiong, "A reputation incentive mechanism of crowd sensing system based on blockchain," in *Artificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part II 6*. Springer, 2020, pp. 695–706.

[23] S. Xia, F. Lin, Z. Chen, C. Tang, Y. Ma, and X. Yu, "A bayesian game based vehicle-to-vehicle electricity trading scheme for blockchain-enabled internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6856–6868, 2020.

[24] B. Zhang, X. Wang, R. Xie, C. Li, H. Zhang, and F. Jiang, "A reputation mechanism based deep reinforcement learning and blockchain to suppress selfish node attack motivation in vehicular ad-hoc network," *Future Generation Computer Systems*, vol. 139, pp. 17–28, 2023.

[25] Z. Ning, S. Sun, X. Wang, L. Guo, S. Guo, X. Hu, B. Hu, and R. Y. Kwok, "Blockchain-enabled intelligent transportation systems: a distributed crowdsensing framework," *IEEE Transactions on Mobile Computing*, vol. 21, no. 12, pp. 4201–4217, 2021.

[26] H. Xiao, N. Sidhu, and B. Christianson, "Guarantor and reputation based trust model for social internet of things," in *2015 International wireless communications and mobile computing conference (IWCMC)*. IEEE, 2015, pp. 600–605.

[27] X. Chen and L. Wang, "A trust evaluation framework using in a vehicular social environment," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2017, pp. 1004–1005.

[28] Y. Yu, S. Liu, L. Guo, P. L. Yeoh, B. Vucetic, and Y. Li, "Crowdr-fbc: A distributed fog-blockchains for mobile crowdsourcing reputation management," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8722–8735, 2020.

[29] H. Lin, S. Garg, J. Hu, G. Kaddoum, M. Peng, and M. S. Hossain, "Blockchain and deep reinforcement learning empowered spatial crowdsourcing in software-defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3755–3764, 2021.

[30] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. H. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1251–1266, 2019.

[31] Y. Xu, E. Yu, Y. Song, F. Tong, Q. Xiang, and L. He, "R-tracing: Consortium blockchain-based vehicle reputation management for resistance to malicious attacks and selfish behaviors," *IEEE Transactions on Vehicular Technology*, 2023.

[32] R. Layard, G. Mayraz, and S. Nickell, "The marginal utility of income," *Journal of Public Economics*, vol. 92, no. 8, pp. 1846–1857, 2008, special Issue: Happiness and Public Economics. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0047272708000248

[33] Z. Huang, J. Zheng, and M. Xiao, "Privacy-enhanced crowdsourcing data trading based on blockchain and stackelberg game," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2021, pp. 621–626.

[34] A. B. MacKenzie and L. A. DaSilva, "Game theory for wireless engineers," *Synthesis Lectures on Communications*, vol. 1, no. 1, pp. 1–86, 2006.

[35] Y. Zhang and M. Van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2140–2148.

[36] A. Alharthi, Q. Ni, R. Jiang, and M. A. Khan, "A computational model for reputation and ensemble-based learning model for prediction of trustworthiness in vehicular ad hoc network," *IEEE Internet of Things Journal*, 2023.

[37] B. Zhang, X. Wang, R. Xie, C. Li, H. Zhang, and F. Jiang, "A reputation mechanism based deep reinforcement learning and blockchain to suppress selfish node attack motivation in vehicular ad-hoc network," *Future Generation Computer Systems*, vol. 139, pp. 17–28, 2023.

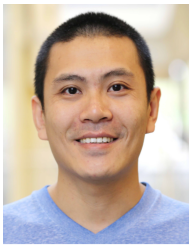
Enze Yu received the B.Eng. degree in information security from Xinjiang University, Urumqi, China, in 2020. He received the master's degree with the School of Cyberspace Security, Southeast Univerity, Nanjing, China, in 2023. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Nanjing University, Nanjing, China. His research interests include edge intelligence and network security.

Yuwei Xu (Member, IEEE) received the B.Eng. degree in information security and the Ph.D. degree in computer science from Nankai University, Tianjin, China, in 2007 and 2012, respectively. He worked as an Assistant Professor with the College of Computer and Control Engineering, from 2012 to 2017, and an Associate Professor with the College of Cyber Science, Nankai University, from 2018 to 2019. He is currently an Associate Professor with the School of Cyber Science and Engineering, Southeast University, Nanjing, China. His research interests include the Internet of Things, future network architecture, and network security.

Gao Lin received the B.Eng. degree in computer science from Central South University, Changsha, China, in 2020. He received the master's degree with the School of Cyberspace Security, Southeast Univerity, Nanjing, China, in 2023. His current research interests include network security and blockchain.

Jie Cao (Student Member,IEEE) will pursue Ph.D. degree in the Department of Electrical and Computer Engineering, Queen's University, Kingston, Canada. He received M.E. degree in CyberspaceSecurity with the School of Cyber Science and Engineering from Southeast University, Nanjing, China in 2024 and BS. degree in Information Security from Xian University of Posts and Telecommunications, Xi'an, China, in 2020. His research interests mainly revolve around the field of cybersecurity and encrypted traffic identification.

Qiao Xiang (Member, IEEE) received the bachelor's degree in information security and the bachelor's degree in economics from Nankai University in 2007, and the master's and Ph.D. degrees in computer science from Wayne State University in 2012 and 2014, respectively. He is a faculty member with Xiamen University. He was previously an Associate Research Scientist with the Department of Computer Science, Yale University. His research interests include software-defined networking, resource discovery and orchestration in collaborative data sciences, interdomain routing, and wireless cyber-physical systems. From 2016 to 2016, he was a Post-Doctoral Fellow with the Department of Computer Science, Yale University. From 2014 to 2015, he was a Post-Doctoral Fellow with the School of Computer Science, McGill University.



Liang He (Senior Member, IEEE) is currently an associate professor at the University of Nebraska-Lincoln (UNL). His research focuses on cyber-physical systems, IoTs, and mobile computing. Before joining UNL, he worked as an assistant professor at the University of Colorado Denver (CU-Denver), a research fellow at the University of Michigan (U-M), a research scientist at Singapore University of Technology and Design (SUTD), and as a research assistant at the University of Victoria (UVic).