Secure Data Sharing over Vehicular Networks Based on Multi-sharding Blockchain

JUNQIN HUANG, LINGHE KONG, JINGWEI WANG, and GUIHAI CHEN, Shanghai Jiao Tong University, China
JIANHUA GAO, Shanghai Normal University, China
GANG HUANG, Zhejiang Lab, China
MUHAMMAD KHURRAM KHAN, King Saud University, Saudi Arabia

Internet of Vehicles (IoV) has become an indispensable technology to bridge vehicles, persons, and infrastructures and is promising to make our cities smarter and more connected. It enables vehicles to exchange vehicular data (e.g., GPS, sensors, and brakes) with different entities nearby. However, sharing these vehicular data over the air raises concerns about identity privacy leakage. Besides, the centralized architecture adopted in existing IoV systems is fragile to single point-of-failure and malicious attacks. With the emergence of blockchain technology, there is the chance to solve these problems due to its features of being tamper-proof, traceability, and decentralization. In this article, we propose a privacy-preserving vehicular data sharing framework based on blockchain. In particular, we design an anonymous and auditable data sharing scheme using Zero-Knowledge Proof (ZKP) technology so as to protect the identity privacy of vehicles while preserving the vehicular data auditability for Trusted Authorities (TAs). In response to high mobility of vehicles, we design an efficient multi-sharding protocol to decrease blockchain communication costs without compromising the blockchain security. We implement a prototype of our framework and conduct extensive experiments and simulations on it. Evaluation and analysis results indicate that our framework can not only strengthen system security and data privacy but also reduce communication complexity by $O(\frac{n\sqrt{m}}{m^2})$ times compared to existing sharding protocols.

CCS Concepts: • **Computer systems organization** → **Embedded and cyber-physical systems**; **Dependable and fault-tolerant systems and networks**; • **Security and privacy** → **Systems security;**

Additional Key Words and Phrases: Internet of Vehicles, blockchain, multi-sharding, scalable, privacy-preserving, zero-knowledge proof

A preliminary version of this article was presented at the 2021 IEEE Global Communications Conference (GLOBECOM).

This work was supported in part by NSFC grants 62141220, 61972253, U1908212, 72061127001, 62172276, 61972254, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, Open Research Projects of Zhejiang Lab No. 2022NL0AB01.

Authors' addresses: J. Huang, L. Kong (corresponding author), J. Wang, and G. Chen, Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, 200240 Shanghai, China; emails: {junqin.huang, linghe.kong, wang159357, chen-gh}@sjtu.edu.cn; J. Gao, Shanghai Normal University, No. 100, Haisi Road, Shanghai, 201412 Shanghai, China; email: jhgao@shnu.edu.cn; G. Huang, Zhejiang Lab Nanhu Headquarters, Kechuang Avenue, Zhongtai Sub-District, Yuhang District, Hangzhou, Zhejiang Province, 311121, China; email: huanggang@zju.edu.cn; M. K. Khan, King Saud University, Riyadh, Po Box 92144, Riyadh 11653 Saudi Arabia; email: mkhurram@ksu.edu.sa.

1550-4859/2024/01-ART31 $15.00
https://doi.org/10.1145/3579035

ACM Reference format:
Junqin Huang, Linghe Kong, Jingwei Wang, Guihai Chen, Jianhua Gao, Gang Huang, and Muhammad Khurram Khan. 2024. Secure Data Sharing over Vehicular Networks Based on Multi-sharding Blockchain. *ACM Trans. Sensor Netw.* 20, 2, Article 31 (January 2024), 23 pages.
https://doi.org/10.1145/3579035

1 INTRODUCTION

Internet of Vehicles (IoVs) can provide real-time communication among different entities, e.g., vehicles, **Road Side Units (RSUs)**, and pedestrians' handheld devices, and aggregate vehicular data from them for safer and smarter transportation management. Due to the superiority of IoV, there are many promising explorations for IoV applications in academia [33], such as autonomous driving, vehicle management, **High-Definition (HD)** map, and big data awareness [15, 26]. Obviously, IoV applications are driven by massive vehicular data, so that securing data privacy, authenticity, and integrity during sharing is a non-negligible part in IoV systems.

However, there are some vulnerabilities in existing IoV systems [21], which will break down the safety of the vehicular data sharing paradigm: *(1) System and data security.* Consider that most IoV systems are built on the centralized architecture, i.e., the **Client-Server (CS)** model, which may suffer from single point-of-failure and malicious attacks [5], such as **Distributed Denial of Service (DDoS)** attacks and Sybil attacks, thereby disabling the functionalities of the whole IoV systems. Furthermore, by tampering with vehicular data stored in the centralized database, vehicles and RSUs can be manipulated by attackers, which could cause traffic chaos. *(2) Identity privacy.* Vehicular data shared over the air can be eavesdropped on and tracked by attackers, who could obtain the identity of vehicles by analyzing vehicular data patterns [31], such as driving track data. The risk of identity privacy disclosure could wear down people's enthusiasm for sharing vehicular data, which hinders the deployment of IoV systems in the real world.

In efforts to solve these problems, Horng et al. [10] devised an identity-based scheme that achieves secure data sharing in vehicular networks. However, their design relies on trusted cloud compute nodes and is vulnerable to a single point of failure. Wei et al. [31] designed a privacy-preserving vehicular communication scheme based on BBS04 group signature, where the group manager acts as a trusted arbiter, but the frequent updating of group members could bring a huge computing burden to the group manager. Yadav et al. [35] proposed a linkable location-based services scheme based on a modified **Linkable Spontaneous Anonymous Group (LSAG)** ring signature scheme, which also needs the trusted parties, i.e., RSUs, as the signature proxies. What's more, such centralized solutions are no longer sufficient to deal with the sophistication of today's cyberattacks; constructing decentralized and zero-trust vehicular networks should be considered as a trending security solution in future IoV systems.

The emergence of blockchain technology has gained considerable attention in recent years. Due to its beneficial characteristics, e.g., decentralization, trusted execution, and tamper resistance, it is promising to solve these problems via the blockchain technology [5, 8, 14]. For example, Chen et al. [6] proposed a quality-driven incentive mechanism based on consortium blockchains for secure data sharing in IoV systems. Su et al. [29] designed a lightweight vehicular blockchain, namely LVBS, for secure data sharing. Even though these solutions improve the system security by a decentralized fabric, they do not consider the identity privacy disclosure of vehicles when applying blockchains. In addition, the limited performance of incumbent blockchains mismatches the demand of high throughput and mobility of IoV systems. Thus, new challenges are also emerging when introducing IoV into the blockchain-based facilities:

- **Conditional identity privacy.** We know that the blockchain is a transparent decentralized ledger, so everyone can obtain blockchain data over networks, which brings threats to the vehicle identity privacy. Even though the vehicle identity can be "anonymized" by blockchain pseudonym accounts, adversaries can reveal the real identity of the vehicle by tracking and analyzing the transactions related to the pseudonym accounts. Thus, the imperfect anonymous scheme of the blockchain will prevent users from contributing their vehicular data in the IoV systems. Besides anonymity, we should retain accountability for **Trusted Authorities (TAs)** to reveal the real identities of malicious nodes and punish them. So, we need a conditional identity privacy-preserving scheme for blockchain-based IoV systems.
- **Performance and scalability.** Since the blockchain involves many complicated technologies, e.g., cryptography and decentralized systems, it is facing huge performance bottlenecks and scalability problems. In particular, the number of consensus nodes can largely impact the convergence speed of the blockchain, thereby affecting the blockchain performance. So, the current blockchain performance and scalability still cannot support the massive data and transactions in the IoV environments.

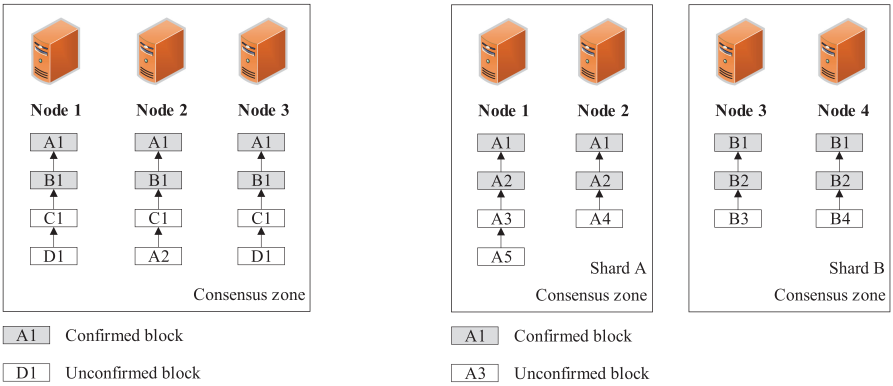
In order to fully address the aforementioned challenges, in this article, we propose a privacy-preserving vehicular data sharing framework atop multi-sharding blockchain. In order to protect the identity privacy of vehicles while retaining the ability of revealing the identity of malicious vehicles for TAs, we design an anonymous and auditable data sharing scheme taking **Zero-Knowledge Proof (ZKP)** technology as basic primitives. So as to bridge the gap between the low performance of blockchains and the high mobility of IoV systems, we design an efficient multi-sharding blockchain protocol for IoV to decrease blockchain communication costs without compromising the blockchain security. Our main contributions are summarized as follows:

- We propose a privacy-preserving vehicular data sharing framework based on blockchain, where we design an anonymous and auditable data sharing scheme for protecting the identity privacy of vehicles while retaining the identity auditability.
- In order to improve the blockchain scalability in IoV systems, we propose an efficient multi-sharding blockchain protocol, which innovatively decouples the shards and the consensus zones, thereby largely reducing communication costs across different shards without sacrificing security.
- To mitigate double-spending attacks in the multi-sharding protocol, we propose a block ordering mechanism to provide the sharding tree the ability to detect transaction conflict. And we provide two optimizations to enhance the security and efficiency of the multi-sharding protocol.
- We implement a proof-of-concept system for the proposed framework and conduct thorough analysis and extensive experiments. The experimental results show that the proposed data sharing framework is secure, privacy preserving, and efficient for IoV systems.

The rest of the article is organized as follows. Section 2 introduces some preliminaries of blockchain sharding and ZKP technologies. Section 3 describes the overview, attack model and design goals, and detailed designs of the proposed framework. We conduct evaluation and analysis of the framework in Sections 4 and 5. Then, we review some related work in Section 6. Finally, a conclusion of this article is given in Section 7, and several future research directions are also discussed.

2 PRELIMINARY

Before we dive into the design of this article, let's briefly introduce the background knowledge of blockchain sharding and zero-knowledge proof.



(a) no-sharding blockchain

(b) sharding blockchain

Fig. 1. The comparison of no-sharding and sharding blockchains.

2.1 Blockchain Sharding

The biggest obstacles to blockchain applications are its performance bottlenecks and scalability issues. Since every decision on the blockchain requires consensus among blockchain nodes, the decentralized consensus protocol has become the main bottleneck of the entire blockchain system. As the number of blockchain nodes increases, the throughput of the system decreases. For this reason, the performance and scalability of traditional blockchains generally cannot meet large-scale scenarios.

Sharding is a term derived from the database field. In a sharding database, data is divided into multiple shards and stored on different server nodes. Every node does not need to store and process all the data. The pressure of data processing is evenly distributed to different nodes, so that the sharding database generally has a better performance and scalability. Similarly, in a sharding blockchain, blockchain nodes and transactions are divided into different shards, and each shard runs the consensus protocol relatively independently. As shown in Figure 1, nodes in the no-sharding blockchain should process all transactions, but nodes in the sharding blockchain only need to handle transactions in their shards. Thus, the sharding blockchain can simultaneously process more transactions compared to the no-sharding one. Assume the shard size is fixed; as the number of shards increases, the performance of the entire system can also be linearly improved. Therefore, a sharding blockchain can accommodate more nodes while maintaining high throughput.

Although the blockchain sharding technologies have great advantages in performance and scalability, its design faces two major design difficulties:

- **Computing power aggregation attacks.** The blockchain sharding technology is a two-edged sword. While improving performance and scalability, blockchain sharding also brings the risk of reducing system security. Taking the sharding blockchain using the proof-of-work consensus protocol as an example, the computing power of honest nodes is dispersed in various shards. Malicious nodes can concentrate resources to attack a certain shard and may easily occupy more than 50% of the computing power of a shard and control the operation of the shard. Similarly, the cost of launching Sybil attacks against sharding blockchain systems using the PBFT consensus protocol is greatly reduced. Therefore, designing the node allocation scheme and consensus mechanism in the sharding systems to prevent computing power aggregation attacks by malicious nodes is the top priority of the blockchain sharding technologies.

- **Cross-shard transactions.** Another big challenge that blockchain sharding technologies need to address is how to handle transactions involving multiple shards. Take the UTXO model adopted by Bitcoin as an example; a transaction often includes multiple inputs and multiple outputs. In a sharding blockchain, these inputs have a probability of coming from different shards; such a transaction is called the cross-shard transaction. The cross-shard transactions need to prove the validity of these inputs across different shards, and the synchronization between different shards cannot rely on consensus protocols. Therefore, designing a powerful and efficient cross-shard communication scheme is also the critical point of the current blockchain sharding technologies.

2.2 Zero-knowledge Proof

ZKP is a cryptographic technique in which a prover proves a proposition to a verifier without revealing any additional information other than "the proposition is true." One of the most popular ZKP schemes is **Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK)** [3]. It is a non-interactive, succinct zero-knowledge proof scheme based on random oracles. The *non-interactive* means that the proof can be completed in the random oracle model without additional interaction; the prover only needs to send a proof message to the verifier to complete verification. The *succinct* means that the length of the proof message sent by the prover is independent of the complexity of the proposition to be proved. It is generally a small constant, and the verification speed of the proof message is fast. The security of ZKP is reflected in the following aspects: *(1) Completeness.* If the proof is true, the proof message from the honest prover to the honest verifier can be verified as true. *(2) Soundness.* If the proof is false, the malicious prover can at most generate a proof message with a negligible probability for the honest verifier to check to be true. *(3) Zero-knowledge.* For the correct proof message, the verifier can only obtain any additional information with a negligible probability except for the information that the proposition is true.

Here we give a general primitive of the zk-SNARK scheme to facilitate our subsequent design:

- $Setup(1^\lambda) \rightarrow PP$. Initialize the system by random numbers and get all public and private parameters.
- $Prover(x, y, PP) \rightarrow u$. The algorithm that generates the proof. x represents the prover's private information, while y represents the prover's public information. The prover runs the $Prover$ algorithm to generate a proof u to prove to the verifier that he or she holds x.
- $Verifier(y, u, PP) \rightarrow 0/1$. Verify the proof algorithm. The verifier runs the $Verifier$ algorithm through y and u to verify whether the prover holds x.

3 VEHICULAR DATA SHARING FRAMEWORK ATOP BLOCKCHAIN

3.1 Overview

Figure 2 shows the overall architecture of the proposed blockchain-based vehicular data sharing framework. In this framework, we consider three types of roles: vehicles, RSUs, and TAs. All of these roles have their own blockchain accounts (public/secret key pairs), which are unique identities used for making transactions (vehicular data sharing) in the blockchain. Because the vehicle must be registered with TAs (e.g., vehicle administration) before it can be used on the road, the vehicular network is a natural permissioned network. So, the vehicular data sharing framework can be constructed on a permissioned blockchain network.

We divide blockchain nodes into full nodes and light nodes according to the size of the role's computing power. RSUs and TAs are the static infrastructures of IoV systems that generally have higher computing power, so that they act as consensus nodes, i.e., full blockchain nodes. The RSUs are the roadside infrastructures (e.g., traffic lights, cameras, street lamps) and responsible

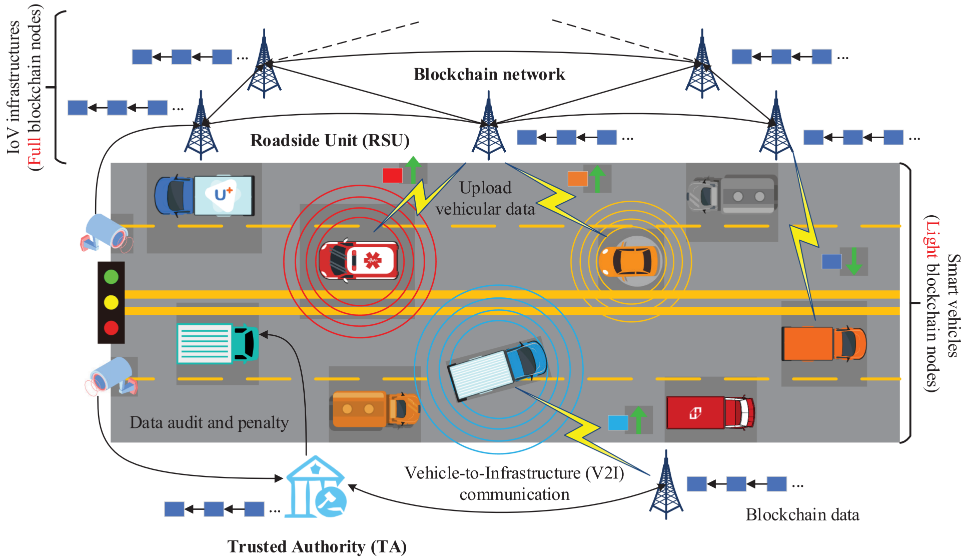


Fig. 2. The overall architecture of blockchain-based vehicular data sharing framework.

for interacting with vehicles. More specific, RSUs collect vehicular data from vehicles and synchronize between other RSUs, and transmit vehicular data to other vehicles nearby, in the form of blockchain transactions. TAs are also full blockchain nodes and responsible for vehicular data audit, as shown in Figure 2. If there are some malicious vehicles that upload bogus data or disguise as other vehicles, TAs have the ability to reveal the real identity of the vehicles and punish them. In comparison, vehicles with high mobility and limited computing resources act as light blockchain nodes. In order to reduce the storage and computation overhead, the light blockchain nodes do not store blockchain data and participate in the process of consensus. They send self-generated vehicular data through RSUs nearby in the form of blockchain transactions and request other vehicles' sharing data from RSUs.

Owing to the decentralization architecture of blockchain, we do not need a trusted centralized server to store or process vehicular data, which strengthens the system reliability. Moreover, vehicular data stored in the blockchain are tamper-proof, which can ensure the integrity of on-chain data. However, on-chain vehicular data may leak the identity privacy of vehicles due to the transparency feature of blockchain. In addition, the limited performance and scalability of incumbent blockchains cannot satisfy the high mobility and throughput demand of IoV systems.

To solve the above two challenges, we first design an anonymous and auditable data sharing scheme for protecting the identity privacy of vehicles while preserving the data auditability for TAs. Then we propose an efficient multi-sharding blockchain protocol, which can improve the performance of blockchains for IoV systems.

3.2 Attack Model and Design Goals

From the point of view of role division, we consider three types of attacks/threats in the proposed framework:

- **Attacks from vehicles.** We assume vehicles are not trusted. Malicious vehicles could report bogus data to the system (*bogus data attack*) or disguise as other honest vehicles (*impersonation attack*).

- **Attacks from RSUs.** RSUs and other infrastructures are assumed to be semi-trusted. Attackers may manipulate a small fraction of RSUs to perform Sybil attack, thereby controlling the network to disrupt traffic and obtain illegal revenue. Attackers could also manipulate RSUs to broadcast false information.
- **Identity privacy disclosure.** Attackers can access all transactions (i.e., vehicular data) due to the transparency of blockchains. Thus, attackers may infer the identity of vehicles by tracking a certain vehicle account (i.e., public key) and analyzing its vehicular data, which causes identity privacy disclosure.

Note that we assume TAs are trusted and secure. Attackers cannot manipulate TAs or steal TAs' secret keys. And we assume that attackers cannot break the cryptographic primitives, including hash inversion attack, digital signature forgery, etc. We conduct security analysis under the above assumptions.

The goals of this article are to design a privacy-protecting but accountable, safe, and efficient data sharing framework for IoV systems. We mainly focus on the following design goals:

- **Decentralization.** Centralized IoV architectures have the disadvantage of single point of failure and opacity. As far as the sharing system of vehicular data is concerned, decentralization and zero-trust are very important. Users can trade data securely and anonymously without trusting any third party.
- **Security.** The ability of the system to function properly under various situations reflects the security and robustness/availability of the system. Our security goal is that the proposed system can execute transactions correctly even under malicious attacks such as double-spending attacks and Sybil attacks. Moreover, the system is robust to DDoS attacks and single points of failure, which means the system service is always available even when some of the nodes are compromised.
- **Privacy.** Identity privacy refers to the anonymity of the vehicles that share data. The identity information of the vehicles that share data should not be disclosed, and the sharing data should not be able to link the real identity of the vehicles.
- **Efficiency.** An important issue to consider is whether the performance and scalability of the blockchain system can keep up with the demand for data sharing in the IoV systems. Therefore, one of the main goals of blockchain-based IoV systems is to achieve high efficiency.

3.3 Multi-sharding Protocol

We observe that an important feature of vehicular data is that it is highly spatial-temporally correlated. In the time dimension, IoV systems generally only focus on data from adjacent times, and stale historical data often lack value. In the spatial dimension, the vehicle basically only pays attention to the nearby vehicular data, and the data that is far away in space is of little value to the vehicle. Therefore, vehicles and RSUs do not need to store all vehicle data, but only care about the data within the surrounding area. Based on this observation, we propose a multi-sharding protocol for blockchain-based IoV systems to improve the performance and scalability without sacrificing the blockchain security.

In the previous blockchain sharding protocols, the consensus nodes would only store the data of one shard and only process the transactions of the corresponding shard. Therefore, when a transaction involves the data of multiple shards, the consensus node needs to communicate with other nodes across shards. Unfortunately, cross-shard communication may need $O(c^2)$ communications in a **two-phase commit (2PC)** scheme [1, 18, 37] to ensure safety (c is the number of nodes in one shard).

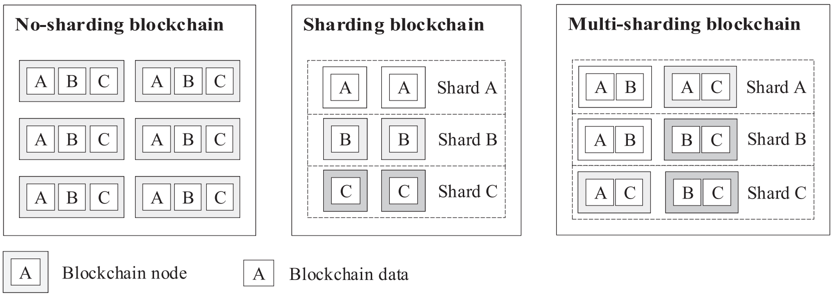


Fig. 3. Consensus groups division in different blockchain sharding protocols.

To mitigate the communication complexity of these sharding protocols, the multi-sharding protocol is designed for improving the performance of cross-shard transactions. The basic idea of the multi-sharding protocol is that by maintaining multiple shards, consensus nodes can directly process cross-shard transactions between these shards. Note that maintaining multiple shards by a consensus node is not the same as maintaining a larger shard. Because different consensus nodes will choose to maintain different shard sets, another feature of multi-sharding is decoupling the concepts of shard and consensus group. In the previous sharding protocols, nodes that maintain a shard will form a consensus group, and they only take part in the consensus process within the shard. In the multi-sharding protocol, a consensus group is the nodes that store the same multiple shards. And nodes in the same consensus group will reach a consensus to determine the execution order and results of transactions in these shards. As shown in Figure 3, for example, the node storing "BC" in shard B will form a consensus group with the node storing "BC" in shard C. Likewise, the node maintaining "AB" in shard A and the node maintaining "AB" in shard B will be in the same consensus group. We can find that shards are no longer equivalent to consensus groups in multi-shard protocols.

The brightest advantage of the multi-sharding protocol is cancelling the 2PC communication process of cross-shard transactions. Assume that a cross-shard transaction contains multiple inputs from shard A and shard B; the nodes in the "AB" consensus group can directly process this cross-shard transaction without additional cross-shard communication, because they simultaneously maintain the blockchain data of shard A and shard B. In this way, the multi-shard protocol adopts a space-for-time strategy, which can effectively reduce the communication complexity of cross-shard transactions and improve the sharding system performance.

Next, we introduce the multi-shard protocol design in detail from the three modules of *shard storage and consensus strategy*, *block ordering mechanism*, and *reconfiguration and data pruning*. For ease of description, we denote an RSU as $r \in R$ and a vehicle as $v \in V$.

3.3.1 *Shard Storage and Consensus Strategy.*

In this module, we design the shard storage and consensus strategy, including the shard division, selection, storage, and consensus process.

Shard division. Since full blockchain nodes such as RSUs naturally belong to different administrative regions geographically, we can divide them into different shards by geographical location. We divide the entire IoV system into A areas according to location, and denote the ith area as $area_i$. Each area contains several RSUs, and vehicles move between different areas. We regard an area as a shard and denote the ith shard as $shard_i$. The number of shards S is obviously equal to A. When a vehicle generates data in the $area_i$, this data will be stored in the corresponding $shard_i$.

Multi-shard selection. After shard division, consensus nodes in the system (including RSUs and TAs) will choose k shards to store, where $k \leq S$. k is a configurable parameter. It can be found that when $k = 1$, the multi-sharding protocol will degrade to the traditional sharding protocols. When $k = S$, the multi-sharding protocol will be the same as the no-sharding blockchain protocols. In addition to the shard where the consensus nodes are located, consensus nodes will randomly store other $k - 1$ shards from the remaining $S - 1$ shards. That is, a consensus node in the $area_i$ will store all the data of the shards

$$\left\{shard_i, C_{S-1}^{k-1}(shard_1, \ldots, shard_{i-1}, shard_{i+1}, \ldots, shard_S)\right\}.$$

For the convenience of description, we take $k = 2$, $S = 32$ as an example for illustration. When $k = 2$, the consensus nodes in the $area_i$ only store the data of two shards, i.e., $shard_i$ and $shard_j$, where $i \neq j$. The $shard_j$ is calculated by the blockchain address $addr$ of the consensus node concatenating a public random number $RAND$:

$$j = TRUNCATE\left(HASH\left(addr||RAND\right), 5\right),$$

where $TRUNCATE(*, 5)$ truncates the first 5 bits of the string and $HASH(*)$ denotes a hash function/random oracle. Since the output of the hash function is random and uniform, the truncated result of j is also uniformly distributed, which makes the multi-shard selection of consensus nodes random and balanced. Due to this verifiable multi-shard selection scheme, the consensus nodes can easily verify if other nodes select and store shards correctly. If not, they will not accept the results from illegal nodes. Therefore, malicious nodes cannot arbitrarily select shards for computing power aggregation attacks. We denote the consensus node set storing the data of $shard_i$ and $shard_j$ as N_{ij}. Note that $N_{ij} \subset N_i \cap N_j$. This shard selection scheme can easily be extended to $k = 3$ or more; just keep truncating the first 5 bits of the rest of the string as indexes for other shards.

Consensus zone. In the multi-sharding protocol, the storage contents of consensus nodes in the same shard are not consistent, so the consensus zone is not in the shard, but the set of nodes that store the same multiple shards constitute a consensus zone. Figure 4 shows the consensus zone of the multi-sharding protocol in the case of two-shard, i.e., $k = 2$. We can see that Node 2 and Node 3 store the data of shard A and shard C at the same time, so the two nodes will run the consensus protocol together to confirm the order and validity of the block data.

Block A1 in Figure 4 represents the first block that stores the data of shard A, while block AB1 represents the first block that stores the cross-shard transactions between shard A and shard B. Since there is a state update in the cross-shard transaction, this transaction should be synchronized in each involved shard. For example, the cross-shard transaction AB1 should be updated in shard A and shard B at the same time. It can be found that the consensus nodes that store both shard A and shard B can actually directly run the consensus protocol to process block AB1, because other nodes in the same consensus zone also store the data of these two shards. So, they do not need to process cross-shard transactions through 2PC communication steps. The nodes outside the consensus zone do not participate in the processing of blocks. These node sets that store both shard A and shard B can be called consensus zone AB. Therefore, by decoupling the consensus zone with the shard, the node can directly process the cross-shard transaction without designing an additional cross-shard communication mechanism.

Without loss of generality, the cross-shard transaction tx_{ij} involved in the $area_i$ and $area_j$ will be processed by the consensus node set N_{ij}. Nodes in N_{ij} will verify the authentication of tx_{ij} and add it to the block $block_{ij}$. Obviously, all transactions in $block_{ij}$ generated by N_{ij} only involve transactions between vehicles in the area $area_i$ and area $area_j$. And this cross-shard transaction processing can easily be extended to $k = 3$ or more.

Blockchain structure. As mentioned above, in the multi-sharding protocol, there are a large number of consensus zones in the system, and the blocks generated by some consensus zones also need to be stored by nodes outside the consensus zone. For example, both nodes in shard A and

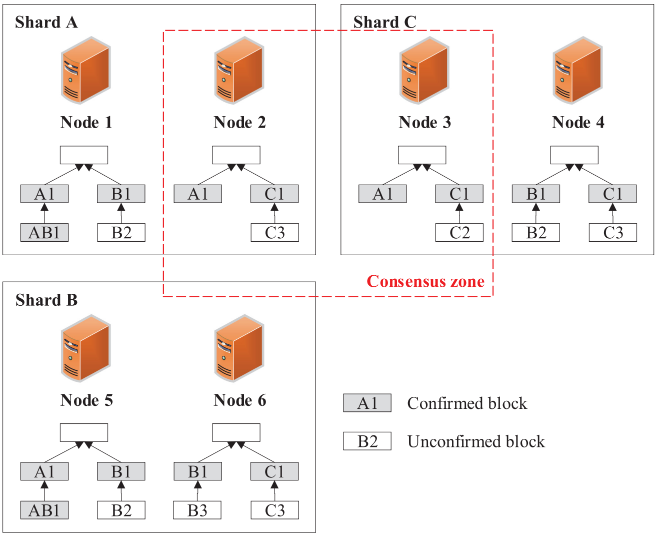


Fig. 4. Shards and consensus zones are decoupled in the multi-sharding protocol.

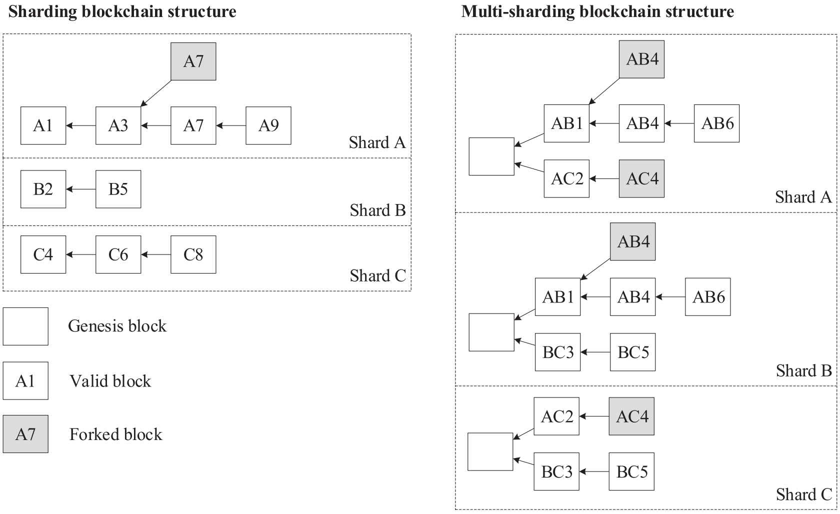


Fig. 5. The comparison of blockchain structure between multi-sharding and existing sharding protocols.

shard B must store blocks generated by the consensus zone AB, even though some of the nodes do not belong to this consensus zone. Under this design, the structure of blockchain data in a shard is no longer a chain, but a tree, as shown in Figure 5. All blocks in a shard have a common ancestor, which is the genesis block. All $block_{ij}$ will form a chain, which is created by N_{ij}, where $1 \leq i, j \leq S$. When $i = j$, it means this chain does not contain cross-shard transactions. Thus, a shard will contain S chains, and each chain is extended by a consensus zone. Take Figure 5 as an example; the shard A should have three chains created by the consensus zones A, AB, AC, respectively.

3.3.2 *Block Ordering Mechanism.* Since the blockchain structure of the shard is a tree, the ability to defend against-spending attacks of the traditional blockchains has disappeared. Let's first

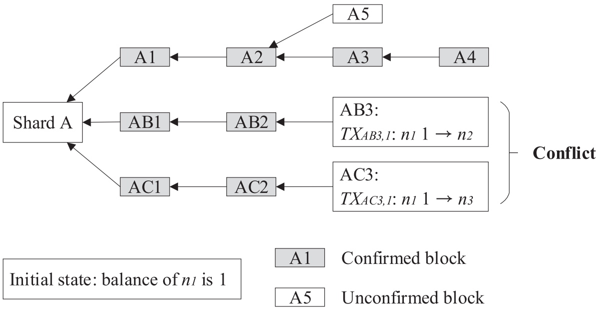

A5
A1
A2
A3
A4
Shard A
AB1
AB2
AB3:
$TX_{AB3,1}$: n_1 1 → n_2
AC1
AC2
AC3:
$TX_{AC3,1}$: n_1 1 → n_3
Conflict
Initial state: balance of n_1 is 1
A1 Confirmed block
A5 Unconfirmed block

Fig. 6. The double-spending example in the multi-sharding protocol.

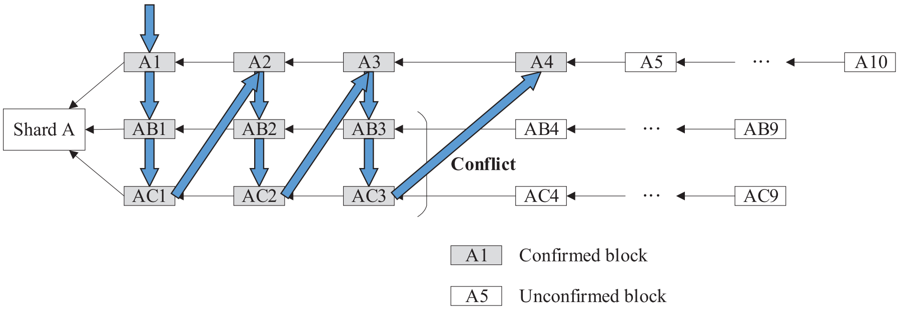
explain this problem through an example. Suppose the vehicle v in the $shard_i$ initiates two transactions at the same time to request the data in the $shard_j$ and $shard_k$, and the price of the target vehicular data is one token and the balance in the account of vehicle v is also one token. Unfortunately, the consensus nodes in N_{ij} and N_{ik} both think that tx_{ij} and tx_{ik} are legal and add these two transactions to their blocks, respectively. Since these two transactions are on different chains of a shard tree, it is impossible to directly roll back the block through transaction conflict. Therefore, an attacker can easily complete a double-spending attack, as shown in Figure 6.

The fundamental reason is that the tree structure in the multi-sharding protocol destroys the timing logic between blocks. Transactions in different blocks may conflict, as shown in Figure 6. However, there is no clear timing mechanism to decide the sequence of different blocks and determine which block is illegal. Therefore, the current multi-sharding protocol design does not solve the problem of transaction order consistency, resulting in double-spending attacks. We propose a block ordering mechanism, which sorts the tree structure in a shard and regenerates it into a chain structure, so as to solve the double-spending and other inconsistencies caused by the tree structure.

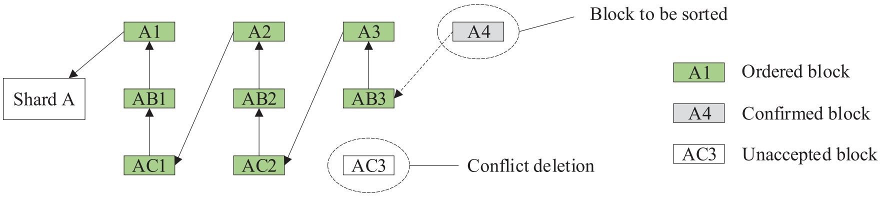
The detailed design is shown in Figure 7. First of all, there are two types of blocks in the system, namely confirmed blocks and unconfirmed blocks, which are distinguished based on the "6 block confirmation" mechanism adopted in Bitcoin. During block sorting, all conflict blocks will be deleted. When sorting, each time a confirmed block is taken from each chain of the sharding tree, it is sorted according to the order of the consensus zone number, and then the process is repeated again and again. If a block that conflicts with a sorted block is encountered, it will be deleted. And if an unsorted block is encountered, it will wait for block confirmation. The multi-sharding protocol only recognizes that the ordered blocks are the final accepted blocks by the system.

3.3.3 Reconfiguration and Data Pruning. In this module, we introduce two optimizations to make the multi-sharding protocol more secure and efficient.

Reconfiguration. To further resist the computing power aggregation attacks of malicious nodes, the consensus nodes will regularly update the stored shards; that is, the system will be reconfigured. Similar to many blockchain sharding protocols (e.g., Rapidchain [36]), we assume that the attacker is slowly adaptive and the protocol process will be divided into multiple time epochs. A slowly adaptive attacker can only destroy a set of complete nodes at the beginning of the protocol or each epoch, but cannot repeatedly destroy nodes within an epoch. Since the consensus of $block_{ij}$ is only completed in N_{ij}, in order to ensure that the computing power of attackers in any consensus group will not exceed 1/2, we need to run the reconfiguration mechanism at the beginning of each epoch to reconfigure the consensus nodes and their stored shards. Specifically, we require RSUs to periodically reconfigure their maintenance shards through the proposed multi-shard selection strategy. We set the public random number $RAND$ as the latest block height, and RSUs compute



(a) block ordering process



(b) block ordering result

Fig. 7. Block ordering mechanism in the multi-sharding protocol.

the new j to select $k-1$ new shards. Since the output of j is random and verifiable, malicious RSUs cannot decide the concrete shards they need to maintain in the next epoch. In this way, malicious RSUs cannot collude with each other in advance to compromise one specific consensus group.

Data pruning. Even though the multi-sharding protocol has decreased the blockchain data volume stored in RSUs and TAs, there is still more and more historical vehicular data in the shard over time. It is not efficient to keep all the historical data like the traditional blockchain. Due to the high temporal selectivity of vehicular data, stale temporal data is actually of little value. Therefore, we prune stale historical blockchain data in the framework to save the storage resources. Specifically, RSUs only retain the metadata (i.e., block headers) of those blocks that have appeared for a long time, and the concrete vehicular data in the block body is deleted to ensure the content stored in the shard will not be too large. The data pruning strategy will not affect the system security, because RSUs still can check the integrity of the blockchain data through the stored metadata. If a malicious RSU tries to modify the historical blockchain data, it needs to change the metadata of historical blocks, which can be easily found and rejected. Considering real IoV scenarios, we define vehicular data of more than 1 day as historical data. In this way, RSU only needs to store the complete block data of the latest day, which can greatly reduce storage overhead. Since TAs are trusted parties and responsible for monitoring the operation of the system, they should still store complete blockchain data for future audit and reference.

3.4 Anonymous and Auditable Data Sharing Scheme

In order to protect the identity privacy of vehicles, ensure the authenticity of vehicular data, and reveal the identities of malicious vehicles, we design an anonymous and auditable privacy-preserving scheme based on ZKP technology for blockchain-based IoV systems.

Considering the requirements of the IoV system, this scheme should satisfy the following objectives: *(1) Anonymity.* Vehicles can know about the authenticity of obtained vehicular data, i.e., the vehicular data is from legal vehicles, but there is no way to reveal the real identities of

Table 1. Description of Symbols Used in the Anonymous and Auditable Data Sharing Scheme

Symbol	Description
(msk, mpk)	The public/secret key pair of TA
(sk, pk)	The public/secret key pair of vehicle
$cert_{pk}$	The certificate for vehicle's public key pk issued by TA
m	The vehicular data to be shared
ek	The ciphertext encrypted with mpk, $ek = Enc(mpk, pk\|\|m)$
u	The zero-knowledge proof generated by $Auth$ primitive
$Auth$	The primitive for generating a proof that data is from a legal vehicle (available for vehicle)
$Verify$	The primitive for verifying if a proof is correct (available for vehicle, RSU, TA)
$Reveal$	The function for decrypting ek to reveal the vehicle identity (available for TA)
$CertGen$	The function for generating $cert_{pk}$ for a vehicle (available for TA)

vehicles through tracking and analyzing vehicular data. *(2) Auditability.* When a vehicle is doing bad things, such as reporting false data, TAs can audit the real identities of malicious vehicles and punish them if necessary. *(3) Non-interactive.* The scheme should be non-interactive, as the communication window between mobile vehicles and RSUs may be short, thereby not supporting multiple rounds of communication. *(4) High computing efficiency.* The computing power of vehicles generally is weak, so the computation cost of vehicular data authenticity verification should be efficient, even negligible, for vehicles.

For the above objectives, we design the primitives of this scheme based on the non-interactive ZKP technology, namely zk-SNARK [9], which has been introduced in Section 2. The basic idea of this scheme is that each vehicle has a public/secret key pair to represent its identity. In order to join the IoV system, the vehicle needs to register its identity information to TAs and get a certificate. When generating a transaction of sharing vehicular data, the vehicle proves the legality of its public/secret key pair and certificate by using the zk-SNARK technique, while other vehicles cannot obtain any identity information of the vehicle. At the same time, the vehicle identity encrypted with the TA's public key is also included in the transaction, so TA can directly decrypt it to obtain the real identity of the vehicle if necessary. We explain the meaning of the symbols used in this section in Table 1.

To realize the design goal of anonymity, we construct two primitives based on zk-SNARK, i.e., $Auth$ and $Verify$, which are described as follows:

- $Auth(m, sk, pk, cert_{pk}, mpk) \rightarrow (m, ek, u)$. $Auth$ primitive is constructed on the zk-SNARK $Prover(x, y, PP)$ function [9]. When a vehicle wants to launch a new transaction in the system, it needs to use the sk,pk,$cert_{pk}$,mpk as inputs to run the $Auth$ primitive. In particular, $Auth$ primitive first calculates $ek = Enc(mpk, pk||m)$, which encrypts the vehicle's public key pk, concatenating message m with mpk. Let $x = (sk, pk, cert_{pk})$ be the private input (i.e., witness) and $y = (m, mpk)$ be the public input. $Auth$ primitive outputs m, ek, and u, which are broadcast to the blockchain network as a transaction. $Auth$ primitive can generate a proof message u to prove that the vehicle has a valid certificate issued by TAs. Due to the *zero-knowledge* guarantee of zk-SNARK, other vehicles learn nothing about the private input x but know if the identity is valid. Besides, TAs can audit the real identity of the vehicle by decrypting ek using msk when necessary. So, $Auth$ primitive can provide an anonymous and auditable proof generation function for IoV systems. We also show the $Auth$ implementation in Listing 1.
- $Verify(m, mpk, ek, u) \rightarrow 0/1$. $Verify$ primitive is built on the zk-SNARK $Verifier(y, u, PP)$ algorithm [9]. When consensus nodes (i.e., RSUs) or other vehicles receive a transaction,

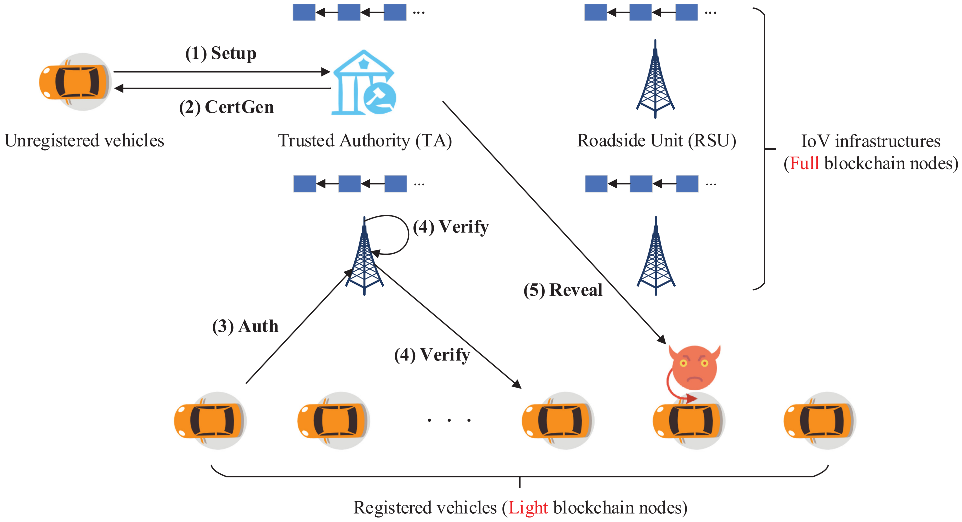


Fig. 8. The workflow of the anonymous and auditable data sharing scheme.

they can run the $Verify$ primitive to verify if the vehicular data in the transaction is from a legal vehicle. The $Verify$ primitive takes $y = (m, mpk, ek)$ and u as inputs and outputs 0/1 to indicate whether the proof is correct. If the output is 1, it means the vehicular data is from a registered vehicle, otherwise 0.

```
def Auth(private sk, private pk, private cert_pk, public
    mpk, public m) -> (public ek, public u):
  // check if (pk, sk) is a correct key pair, prove
      ownership of sk
  require(proofOfOwnership(pk, sk) == 1)
  // verify the validity of cert_pk using mpk
  require(certverify(pk, cert_pk, mpk) == 1)
  // encrypt pk||m with mpk for future revealing
  ek = Enc(mpk, pk||m)
  // generate a succinct proof message
  u = Prover(sk, pk, cert_pk, mpk, m, ek)
  return (ek, u)
```

Listing 1. *Auth* primitive implementation.

Based on the above constructed primitives, the workflow of the anonymous and auditable data sharing scheme is shown in Figure 8, and we describe it as follows:

(1) **Setup.** During the system initialization process, the TA generates a public/secret key pair (mpk, msk). mpk is known to all system participants as built-in information in the system. When a new vehicle joins the system, it first generates a public/secret key pair (pk, sk) and registers its identity pk in the TA.
(2) **CertGen.** The TA will use its secret key msk to generate a certificate $cert_{pk}$ for the vehicle to prove the legality of the vehicle. All vehicles need to register in the TA before sharing vehicular data in the IoV system; otherwise their transactions will be discarded by other nodes.

(3) **Auth.** Before a vehicle shares its vehicular data m, it will execute $Auth$ primitive to generate a proof u that proves the vehicular data is from a registered vehicle. Then, the vehicle takes the output (m, u, ek) as a transaction and sends it to the nearby RSU. It is noted that $ek = Enc(mpk, pk||m)$ varies for different vehicular data, so the identity of vehicles will not be linked by ek.

(4) **Verify.** When an RSU receives a new transaction, it executes $Verify$ primitive to verify that the transaction is from a registered vehicle. If the $Verify$ outputs 1, it means the transaction is legal. The RSU will broadcast it to other consensus nodes through the gossip protocol. If not, it will discard this illegal transaction immediately. When a vehicle obtains vehicular data from nearby RSUs, it can execute $Verify$ primitive to validate if the vehicular data source is legal.

(5) **Reveal.** The reveal process is a decryption function, denoted as $Reveal(ek, msk) \rightarrow pk$. When the TA finds that there are some abnormal vehicular data in the IoV system, it can decrypt the ek in the corresponding transaction and execute the $Reveal$ function to decrypt ek using its secret key msk. Then, the function will output the vehicle's public key pk, thereby revealing the real identity of the vehicle. So, only the TA has the ability to reveal the identity of vehicles and punish them by legal means.

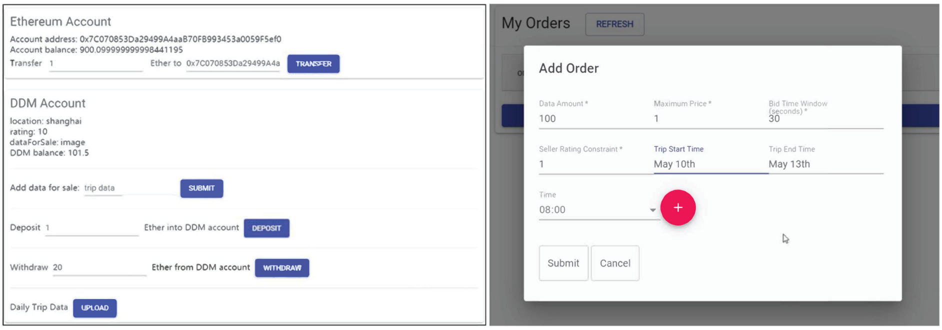
4 SECURITY ANALYSIS

To demonstrate the security of the proposed framework, we give a sketch analysis against three attack types described in Section 3.2:

Against attacks from vehicles. We assume that malicious vehicles could upload forged data or pretend to be other honest vehicles. To defend such attacks, we need TAs to have the ability to audit data and reveal the real identity of malicious vehicles. TA's data auditing and punishment capabilities can punish fake data attacks and impersonation attacks from vehicles. We assume that vehicles will not leak their secret keys. Then if a malicious vehicle wants to generate a transaction that "looks legitimate," it must use its own public/secret key pair and certificate signed by TAs to perform $Auth$ calculations to generate a valid proof for RSUs and other vehicles to verify. Due to the *completeness* and *soundness* properties of ZKP technology, TAs and other nodes can ensure the validity of the pseudo identity information ek in the transaction, so an uncertified malicious vehicle cannot launch impersonation attacks to pretend to be certified ones. What's more, the ek is encrypted by TA's public key; when TA finds that there exists wrong/fake vehicular data sharing in the system, it can reveal the real identity pk of the corresponding vehicle by decrypting ek and further punish its malicious behaviors. So, the proposed framework can well defend attacks from malicious vehicles.

Against attacks from RSUs. We assume that manipulated RSUs could launch Sybil attacks and possible double-spending attacks. Since RSUs should register in the TA before joining the system, every RSU only has a legitimate public/secret key pair. So, if an attacker tries to launch Sybil attacks to compromise the whole system, it generally needs to collude with more than half of the RSUs in a shard, which is extremely difficult when the number of RSUs is large enough. Besides, we defend possible double-spending attacks through the proposed block ordering mechanism, which transforms the tree-structure of the blockchain into the chain-structure, thereby gaining the conflict detection ability in the multi-sharding protocol. Here we prove the correctness of the block ordering mechanism by contradiction:

Proof. We assume that the block ordering result of different nodes is inconsistent. As described in Section 3.3.2, only the confirmed blocks participate in the sorting. According to the basic concept of blockchain, every honest node should have the same view for confirmed blocks; i.e., they have



Ethereum Account

Account address: 0x7C070853Da29499A4aaB70FB993453a0059F5ef0
Account balance: 900.099999999998441195
Transfer 1 Ether to 0x7C070853Da29499A4a

DDM Account

location: shanghai
rating: 10
dataForSale: image
DDM balance: 101.5

Add data for sale: trip data

Deposit 1 Ether into DDM account

Withdraw 20 Ether from DDM account

Daily Trip Data

Add Order

Data Amount * 100
Maximum Price * 1
Bid Time Window (seconds) * 30
Seller Rating Constraint * 1
Trip Start Time May 10th
Trip End Time May 13th
Time 08:00

Fig. 9. The client interface of light blockchain nodes (i.e., vehicles).

the same tree-structure of blockchain in the same shard. Besides, the block ordering mechanism orders the confirmed blocks according to the order of the consensus zone number. Given the same tree-structure of blockchain and the same block ordering rule, every honest node should obtain the consistent block ordering result, which contradicts the assumption. □

Therefore, the block ordering mechanism can ensure that the order of blocks obtained by honest nodes is consistent. Given a consistent order of blocks, honest nodes can easily tell the double-spending transactions by detecting conflicts.

Against identity privacy disclosure. Vehicles use the *Auth* primitive to hide their real identities before sharing vehicular data and to prove the validity of their identities. To unlink the real vehicle identity and vehicular data, we encrypt the vehicle's public key *pk* and vehicular data *m* with TA's public key *mpk* and take the encrypted string *ek* as the pseudo identity. Since *ek* is different for different vehicular data, the vehicle cannot be traced by analyzing its vehicular data and its real identity will not be revealed. Due to the *zero-knowledge* property of ZKP technology [9], the RSUs and vehicles cannot know any identity information about the vehicular data uploader, except the validity of its pseudo identity *ek*. Thus, the *unlinkability* between the vehicle identity and its vehicular data holds.

5 EVALUATION

5.1 Experiment Setup

Implementation. We implement a prototype of the proposed vehicular data sharing framework. The implementation of the multi-sharding protocol is based on Ethereum. We modify the consensus and storage process of the Ethereum source code. For the simplicity and flexibility of implementation, the prototype uses the proof-of-authority consensus protocol in the private test net to replace the proof-of-work consensus protocol in the Ethereum main net. To simulate data transactions from vehicles, we implement a client interface for light blockchain nodes. The client interface supports transfer, vehicular data upload, and purchase functions, as shown in Figure 9. Besides, we choose *Groth16* [9] as our underlying zk-SNARK scheme and leverage the jsnark library to implement the anonymous and accountable data sharing scheme. The prototype is open source and available at https://github.com/imtypist/AAVBvanet.

Experiment settings. In order to truly reflect the performance of our framework, we use a real-time traffic dataset, SUVnet [12], which was collected from over 4,000 taxis in Shanghai, China, including the latitude and longitude position, speed, heading, and other information of taxis, as shown in Figure 10. We also use **Simulation of Urban MObility (SUMO)** to simulate the continuous change in space and discrete change in time of the IoV system to reflect the real performance

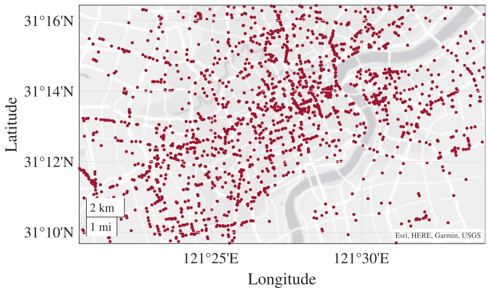


Fig. 10. Visualization of the SUVnet on March 1, 2007, at 09:00. Every dot represents a taxi.

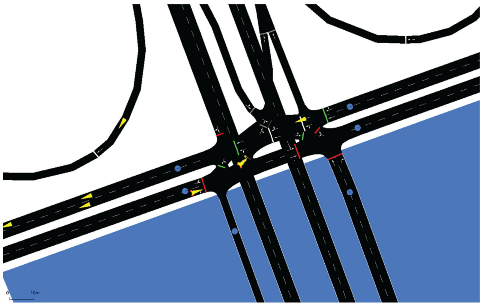


Fig. 11. The SUMO software simulation of the vehicular data sharing system.

Table 2. The Comparison of Multi-sharding Protocol and Existing Blockchain Sharding Protocols

	No-sharding	Sharding	Multi-sharding*
Storage redundancy	$O(n)$	$O(\frac{n}{m})$	$O(\frac{n}{\sqrt{m}})$
Bandwidth consumption	$O(n)$	$O(\frac{n^2}{m^2})$	$O(\frac{n}{\sqrt{m}})$
Throughput	$O(1)$	$O(m)$	$O(m)$
Security	n	$\frac{n}{m}$	$\frac{n}{m}$

n: the total number of consensus nodes, m: the number of shards, $\frac{n}{m} = c$: the size of a shard.
*We set the number of shards to $m' = \sqrt{m}$ in the multi-sharding protocol for gaining the same security level and throughput improvement as previous sharding schemes.

of the multi-sharding protocol, as shown in Figure 11. We run full blockchain nodes (i.e., RSUs and TA) on PC with Intel CPU i7-7700, 16 GB RAM. To simulate the limited computing power of vehicles, we run light blockchain nodes on VirtualBox with two-core CPU, 4 GB RAM.

5.2 Multi-sharding Protocol

5.2.1 Theoretical Analysis. To demonstrate the highlights of the multi-sharding protocol, we provide an informal theoretical comparison of the multi-sharding protocol with existing no-sharding and sharding blockchain protocols; see Table 2. The existing sharding protocols are mostly based on the 2PC cross-shard communication protocol, such as Omniledger [18], CycLedger [37], or Chainspace [1], which have similar settings on consensus approach, cross-shard communication, thereby achieving similar performance. The cross-shard communication protocol adopted in RapidChain [36] only supports simple currency transfers, which cannot be applied in IoV systems, so we do not compare with it here.

We define four evaluation indicators to compare the performance of our solution and previous methods. The *storage redundancy* represents the storage resources consumed by a transaction. The *bandwidth consumption* represents the communication complexity for a transaction, which characterizes the efficiency of cross-shard communication. For simplicity, we assume that all transactions in the system are cross-shard transactions (there is no difference between cross-shard transactions and in-shard transactions for no-sharding blockchains), and these transactions only involve two shards. The *throughput* represents the theoretical throughput magnitude under different schemes. The *security* represents the number of nodes in a consensus zone, which can qualitatively describe the possibility of malicious nodes successfully manipulating a consensus zone. The number of nodes in a consensus zone is larger; the security is stronger.

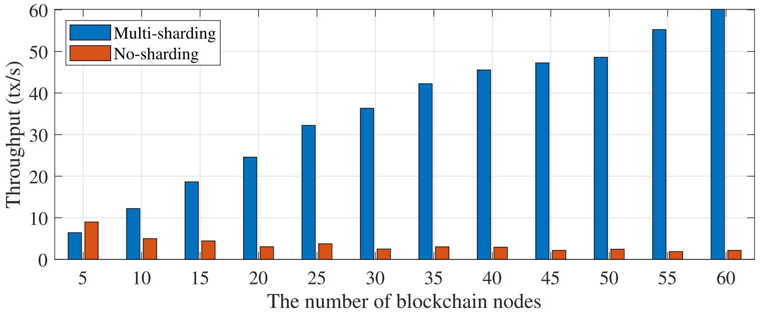


Fig. 12. Throughput comparison between multi-sharding and no-sharding blockchain protocols.

For the previous sharding protocols that use the 2PC cross-shard communication scheme, the storage redundancy is $O(\frac{n}{m})$ and the security is $\frac{n}{m}$. As one shard can be seen as a relatively independent consensus zone, m shards in the sharding protocol can theoretically gain m times throughput of the no-sharding protocol. Most importantly, the 2PC cross-shard communication scheme costs $O(c^2)$ rounds of interactions, which is a major bottleneck of the sharding protocols.

The multi-sharding protocol has $O(\frac{n}{m})$ storage redundancy, $O(\frac{n}{m})$ bandwidth consumption, $O(m^2)$ throughput, and $\frac{n}{m^2}$ security. Security is a critical problem that cannot be ignored in blockchains. In the multi-sharding protocol, we can reduce the number of shards to achieve the same security level as the previous sharding protocols, i.e., setting the number of shards in the multi-sharding to $m' = \sqrt{m}$. From Table 2, we can find that the multi-sharding protocol has the same throughput and security compared with previous sharding protocols. While the bandwidth consumption of the multi-sharding protocol is better than the existing sharding protocols (reducing $O(\frac{n\sqrt{m}}{m^2})$ times), it has a higher cost of storage resources. But with the ever-decreasing cost of storage, trading space for time is a good option.

5.2.2 Performance Evaluation. We conduct simulation experiments on no-sharding, sharding, and multi-sharding protocols to evaluate the correction of the above analysis. The experiment considers 1,000 randomly generated transfer transactions, each with a length of about 400 bytes. In the experiment, transactions are added to the implemented system from time to time, and the transaction latency and communication consumption of the system are recorded. Through repeated tests, we discuss the throughput and bandwidth consumption of the system under different numbers of consensus nodes and different shard sizes.

Figure 12 shows the throughput comparison between the multi-sharding and no-sharding protocol when setting each shard containing five blockchain nodes. We can observe that with the increasing number of the consensus nodes, the throughput of the multi-sharding blockchain protocol will near-linearly increase. The reason is that the size of consensus zones in the multi-sharding protocol is fixed, and different consensus zones are parallel. Thus, the overall system throughput is linear to the number of consensus zones (i.e., shards) and the system has a good scalability. In comparison, the throughput of the no-sharding blockchain system has a slight decrease with the increasing blockchain nodes, because it will consume more time to reach consensus among a larger size of consensus zone.

Figure 13 shows the bandwidth consumption comparison between the sharding and multi-sharding protocol. In the experiment, we fix the number of shards and adjust the number of blockchain nodes. We can observe that the bandwidth consumption of the sharding protocols will increase non-linearly and sharply with the number of nodes, and finally the throughput of the

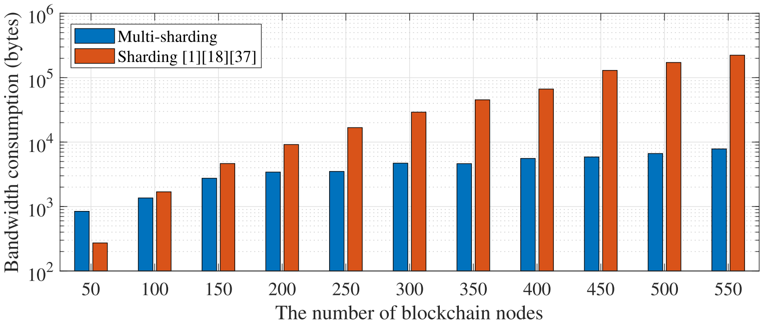


Fig. 13. Bandwidth consumption comparison between sharding and multi-sharding blockchain protocols (y-axis is log-scale).

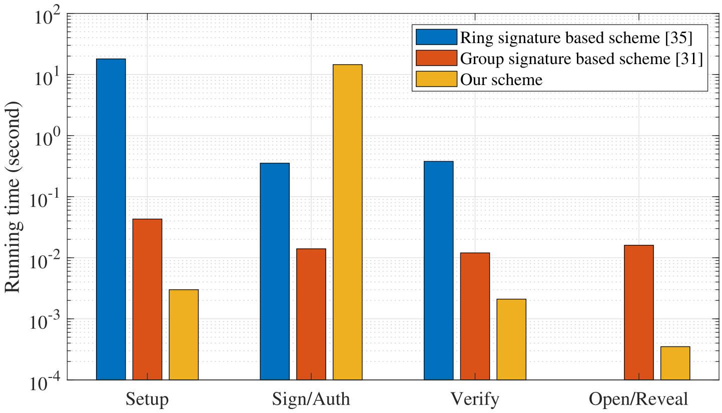


Fig. 14. Running time comparison of privacy-preserving schemes between ours and other solutions (y-axis is log-scale).

system will also be constrained by the limited bandwidth of the consensus nodes. In comparison, the multi-sharding protocol performs well and can achieve near-linear bandwidth consumption growth as the number of nodes increases. Thus, the multi-sharding protocol has a better scalability due to the lower communication complexity, which is more practical to bandwidth-sensitive IoV scenes.

5.3 Anonymous and Auditable Data Sharing Scheme

Aside from the security goals (i.e., anonymity and auditability) mentioned in Section 3.4, we also have two performance requirements: non-interactive and high computing efficiency. The demand of reducing communication interactions (i.e., non-interactive) has been naturally achieved by zk-SNARK-based primitives. For evaluating the computing efficiency of the proposed scheme, we compare the running time of our scheme and two related solutions, which are respectively the ring-signature-based [35] and the group-signature-based [31] schemes. We implement the group signature scheme based on BBS04 [4] and the ring signature scheme based on LSAG [20].

The experimental results are shown in Figure 14. The *Setup* step (corresponding to the *CertGen* algorithm in our scheme) is executed only once for each newly added vehicle. For the ring signature and group signature schemes, the execution time of this step increases with the shard size, while our scheme is not affected by the shard size and outperforms the other two solutions. We set the shard size as 32 in the experiment. We observe that our scheme is about 10x faster than the group signature scheme in the *Setup* phase. If the shard size increases, the performance gap between them will be larger. Once the system is set up, vehicles need to run the *Sign/Auth* authentication function before sending data transactions. Our scheme takes about 14 seconds to generate vehicle identification proofs, while the ring-signature-based scheme only needs 0.35 seconds to sign a message. Since our scheme needs to compute multiple evaluations of polynomials in the *Auth* step, our scheme takes longer than the other two *Sign*-based schemes. Fortunately, this step is only performed by the vehicular data owner and generally conducted for every half minute or longer in real IoV scenarios, so a few seconds' execution time is relatively acceptable. Notably, our scheme achieves the best performance in the verification authentication process, with a runtime 5x smaller than related solutions, about 2.1 ms. Since *Verify* will be performed by all consensus nodes and other vehicles, this step may be performed hundreds or thousands of times for each data transaction, which greatly affects the performance of the scheme. Therefore, an efficient performance of *Verify* is particularly important. As for *Open/Reveal*, the ring-signature-based scheme does not have the accountable ability, so it is not applicable in this operation. We observe that our scheme is about 45x faster than the group-signature-based scheme in the *Open/Reveal* step and only takes 0.35 ms.

6 RELATED WORK

Recent advances [6, 29] have been devoted to designing a secure data sharing framework based on blockchain for IoV systems. However, there are still two challenges to be solved: *privacy disclosure* and *performance bottleneck*. We briefly review related solutions from these two aspects.

6.1 Blockchain Privacy-preserving Schemes

Zerocoin [23] is a cryptographic extension of Bitcoin that addresses identity privacy concerns by separating transactions from payment sources without introducing new trusted parties. However, Zerocash [2] pointed out that transactions in Zerocoin still expose the target and amount of the payment, and it provides a more efficient scheme to enforce its privacy. BITE [22] leverages **Trusted Execution Environment (TEE)** hardware, e.g., Intel SGX, to protect the privacy of Bitcoin lightweight clients.

Besides Bitcoin, other blockchain systems are also considering improving their privacy protection capabilities. For example, Monero [25] adopts RingCT [30] to achieve privacy. RingCT is a confidential transaction technology based on ring signature, which is a method to hide the value of transactions. Zcash is a cryptocurrency that is considered to have the strongest guarantee of anonymity. However, an analytical work on Zcash [17] found that its anonymity set can be significantly reduced by developing simple heuristics based on identifiable usage patterns. FabZK [16] is a privacy protection and auditable extension for Hyperledger Fabric, an open source blockchain software community.

However, the above solutions only support simple transfer operations and cannot adapt to more complex application scenarios, such as IoV. To meet this requirement, Kosba et al. [19] proposed a transaction privacy protection scheme based on zero-knowledge proofs that can support smart contracts. Sterling [13] is a blockchain-based privacy-preserving data market system and allows data providers to express constraints such as pricing and differential privacy.

Even though there are many privacy-preserving schemes proposed for blockchains, some studies revealed the incompatibility between existing cryptography technology and blockchain. For example, Ni et al. [24] demonstrated that implementing ring signature schemes in privacy-preserving blockchain systems may be vulnerable to "chain reactions." DIV [34] found that zero-knowledge set membership proofs could incur a huge time and space cost in the case of dynamic membership changes. Thus, it is non-trivial to design a proper privacy protection scheme for blockchain-based IoV systems.

6.2 Blockchain Performance and Scalability Improvement

Blockchain sharding [7] is one of the most popular solutions to improve the performance and scalability of blockchains [11, 32], and there are many researchers focusing on designing an efficient sharding protocol. For example, Zamani et al. [36] proposed RapidChain, the first sharding-based public blockchain protocol that achieves complete sharding of the communication, computation, and storage overhead. Kokoris-Kogias et al. [18] designed an efficient cross-shard commit protocol that atomically handles transactions affecting multiple shards. Zhang et al. [37] presented CycLedger, a scalable and secure parallel protocol for distributed ledger via sharding. However, in these works, the processing of cross-shard transactions is mostly based on the variants of the two-phase commit protocol, and the complexity of cross-shard communication is still quadratic. Besides, these works did not consider the high mobility of vehicles, so they are not suitable for IoV systems where cross-shard transactions happen frequently.

Off-chain-based solutions are also widely adopted in blockchain performance and scalability improvement. The lightning network [27] is an expansion scheme proposed by the Bitcoin community. It is a layer 2 payment protocol that works on Bitcoin. Its main working principle is to put a large number of transactions executed off-chain, and the blockchain is only responsible for storing the results. The security of the lightning network relies on **Revocable Sequence Maturity Contracts (RSMCs)** to solve off-chain confirmation problems and **Hash Time Lock Contracts (HTLCs)** to implement micro-payment channels, but the operations that can be supported at present are still very limited. Side-chain and cross-chain technologies [28] are also common off-chain-based solutions, but so far there is still no good scheme to fully solve the synchronization and authentication problems of these technologies.

7 CONCLUSION AND FUTURE DIRECTIONS

In this article, we propose a privacy-preserving vehicular data sharing framework atop multi-sharding blockchain. First, we design an anonymous and auditable data sharing scheme based on ZKP for protecting the identity privacy of vehicles while retaining conditional auditability. Second, we propose an efficient multi-sharding blockchain protocol, which can achieve lower communication complexity compared to the existing sharding protocols and is more practical for IoV systems. Evaluation and analysis results indicate that our framework can efficiently strengthen the system security and protect the identity privacy.

This article is a preliminary exploration of a new method for blockchain scalability solutions in the IoV scenario, and there are still some aspects that can be followed up for research and exploration. For example, the tree-structured block sorting mechanism in the multi-sharding protocol may be unfair. As the design, the blocks with the higher consensus zone numbers have the sorting priority. When there are two conflicting transactions, the transaction with the higher consensus zone number is always more likely to be confirmed, which may be used for malicious attacks. Thus, enhancing the block sorting mechanism can be valuable future work.

REFERENCES

[1] Mustafa Al-Bassam, Alberto Sonnino, Shehar Bano, Dave Hrycyszyn, and George Danezis. 2018. Chainspace: A sharded smart contracts platform. In *Network and Distributed System Security Symposium (NDSS'18)*. ISOC.

[2] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized anonymous payments from Bitcoin. In *IEEE Symposium on Security and Privacy (S&P'14)*. 459–474.

[3] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. 2012. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS'12)*. Association for Computing Machinery, New York, NY, 326–349.

[4] Dan Boneh, Xavier Boyen, and Hovav Shacham. 2004. Short group signatures. In *Annual International Cryptology Conference (CRYPTO'04)*. Springer, 41–55.

[5] Sheng Cao, Sixuan Dang, Xiaojiang Du, Mohsen Guizani, Xiaosong Zhang, and Xiaoming Huang. 2020. An electric vehicle charging reservation approach based on blockchain. In *IEEE Global Communications Conference (GLOBECOM'20)*.

[6] Wuhui Chen, Yufei Chen, Xu Chen, and Zibin Zheng. 2020. Toward secure data sharing for the IoV: A quality-driven incentive mechanism with on-chain and off-chain guarantees. *IEEE Internet of Things Journal* 7, 3 (2020), 1625–1640.

[7] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. 2016. On scaling decentralized blockchains. In *Financial Cryptography and Data Security (FC'16)*. Springer.

[8] Wenbin Dong, Yang Li, Ronghui Hou, Xixiang Lv, Hui Li, and Bo Sun. 2019. A blockchain-based hierarchical reputation management scheme in vehicular network. In *IEEE Global Communications Conference (GLOBECOM'19)*.

[9] Jens Groth. 2016. On the size of pairing-based non-interactive arguments. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'16)*. Springer.

[10] Shi-Jinn Horng, Cheng-Chung Lu, and Wanlei Zhou. 2020. An identity-based and revocable data-sharing scheme in VANETs. *IEEE Transactions on Vehicular Technology* 69, 12 (2020), 15933–15946.

[11] Huawei Huang, Wei Kong, Sicong Zhou, Zibin Zheng, and Song Guo. 2021. A survey of state-of-the-art on blockchains: Theories, modelings, and tools. *ACM Comput. Surv.* 54, 2, Article 44 (March 2021), 42 pages.

[12] Hong-Yu Huang, Pei-En Luo, Minglu Li, Da Li, Xu Li, Wei Shu, and Min-You Wu. 2007. Performance evaluation of SUVnet with real-time traffic data. *IEEE Transactions on Vehicular Technology* 56, 6 (2007), 3381–3396.

[13] Nick Hynes, David Dao, David Yan, Raymond Cheng, and Dawn Song. 2018. A demonstration of sterling: A privacy-preserving data marketplace. *Proc. VLDB Endow.* 11, 12 (Aug. 2018), 2086–2089.

[14] Tigang Jiang, Hua Fang, and Honggang Wang. 2019. Blockchain-based Internet of Vehicles: Distributed network architecture and performance analysis. *IEEE Internet of Things Journal* 6, 3 (2019), 4640–4649.

[15] Zhenzhen Jiao, Hui Ding, Meimei Dang, Rui Tian, and Baoxian Zhang. 2016. Predictive big data collection in vehicular networks: A software defined networking based approach. In *IEEE Global Communications Conference (GLOBECOM'16)*.

[16] Hui Kang, Ting Dai, Nerla Jean-Louis, Shu Tao, and Xiaohui Gu. 2019. FabZK: Supporting privacy-preserving, auditable smart contracts in hyperledger fabric. In *Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'19)*. 543–555.

[17] George Kappos, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. 2018. An empirical analysis of anonymity in Zcash. In *USENIX Security Symposium (Security'18)*. USENIX Association, Baltimore, MD, 463–477.

[18] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. 2018. OmniLedger: A secure, scale-out, decentralized ledger via sharding. In *IEEE Symposium on Security and Privacy (S&P'18)*.

[19] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *IEEE Symposium on Security and Privacy (S&P'16)*. 839–858.

[20] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. 2004. Linkable spontaneous anonymous group signature for ad hoc groups. In *Australasian Conference on Information Security and Privacy (ACISP'04)*. Springer, 325–335.

[21] Zhaojun Lu, Gang Qu, and Zhenglin Liu. 2019. A survey on recent advances in vehicular network security, trust, and privacy. *IEEE Transactions on Intelligent Transportation Systems* 20, 2 (2019), 760–776.

[22] Sinisa Matetic, Karl Wüst, Moritz Schneider, Kari Kostiainen, Ghassan Karame, and Srdjan Capkun. 2019. BITE: Bitcoin lightweight client privacy using trusted execution. In *USENIX Security Symposium (Security'19)*. USENIX Association, Santa Clara, CA, 783–800.

[23] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. 2013. Zerocoin: Anonymous distributed E-Cash from Bitcoin. In *IEEE Symposium on Security and Privacy (S&P'13)*. 397–411.

[24] Wangze Ni, Peng Cheng, Lei Chen, and Xuemin Lin. 2021. When the recursive diversity anonymity meets the ring signature. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD'21)*. Association for Computing Machinery, New York, NY, 1359–1371.

[25] Shen Noether. 2015. Ring Signature Confidential Transactions for Monero. Cryptology ePrint Archive, Paper 2015/1098. (2015). https://eprint.iacr.org/2015/1098.

[26] Ange Ouya, Blanca Martinez De Aragon, Cécile Bouette, Guillaume Habault, Nicolas Montavont, and Georgios Z. Papadopoulos. 2017. An efficient electric vehicle charging architecture based on LoRa communication. In *IEEE International Conference on Smart Grid Communications (SmartGridComm'17)*.

[27] Joseph Poon and Thaddeus Dryja. 2016. The Bitcoin Lightning Network: Scalable Off-chain Instant Payments. (2016). https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf.

[28] Amritraj Singh, Kelly Click, Reza M. Parizi, Qi Zhang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2020. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications* 149 (2020), 102471.

[29] Zhou Su, Yuntao Wang, Qichao Xu, and Ning Zhang. 2022. LVBS: Lightweight vehicular blockchain for secure data sharing in disaster rescue. *IEEE Transactions on Dependable and Secure Computing* 19, 1 (2022), 19–32.

[30] Shi-Feng Sun, Man Ho Au, Joseph K. Liu, and Tsz Hon Yuen. 2017. RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero. In *European Symposium on Research in Computer Security (ESORICS'17)*. Springer International Publishing, Cham, 456–474.

[31] Lingbo Wei, Jianwei Liu, and Tingge Zhu. 2011. On a group signature scheme supporting batch verification for vehicular networks. In *International Conference on Multimedia Information Networking and Security (MINES'11)*.

[32] Junfeng Xie, F. Richard Yu, Tao Huang, Renchao Xie, Jiang Liu, and Yunjie Liu. 2019. A survey on the scalability of blockchain systems. *IEEE Network* 33, 5 (2019), 166–173.

[33] Wenchao Xu, Haibo Zhou, Nan Cheng, Feng Lyu, Weisen Shi, Jiayin Chen, and Xuemin Shen. 2018. Internet of vehicles in big data era. *IEEE CAA Journal of Automatica Sinica* 5, 1 (2018), 19–35.

[34] Zihuan Xu and Lei Chen. 2021. DIV: Resolving the dynamic issues of zero-knowledge set membership proof in the blockchain. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD'21)*. Association for Computing Machinery, New York, NY, 2036–2048.

[35] Vijay Kumar Yadav, Shekhar Verma, and Subramanian Venkatesan. 2022. Linkable privacy-preserving scheme for location-based services. *IEEE Transactions on Intelligent Transportation Systems* 23, 7 (2022), 7998–8012.

[36] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. 2018. RapidChain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*.

[37] M. Zhang, J. Li, Z. Chen, H. Chen, and X. Deng. 2020. CycLedger: A scalable and secure parallel protocol for distributed ledger via sharding. In *IEEE International Parallel and Distributed Processing Symposium (IPDPS'20)*.

Received 30 August 2022; revised 6 December 2022; accepted 27 December 2022