Secure Data Sharing for Consortium Blockchain-Enabled Vehicular Social Networks

Mingming Cui, Dezhi Han, *Senior Member, IEEE*, Han Liu, Kuan-Ching Li, *Senior Member, IEEE,* Mingdong Tang, *Member, IEEE*, Chin-Chen Chang, *Fellow, IEEE,* Ferheen Ayaz, *Graduate Student Member, IEEE,* Zhengguo Sheng, *Senior Member, IEEE,* and Yong Liang Guan, *Senior Member, IEEE*

***Abstract*—Data sharing in Vehicular Social Networks (VSNs) is an essential road service that assists vehicle driving and promotes intelligent transportation applications. In VSNs, vehicles regularly collect and upload valuable data to share with other vehicles. Data encryption can be employed during data uploading and sharing to prevent malicious tampering and privacy disclosure. However, existing data-sharing schemes lack security, have high overhead in obtaining decrypted data, and show low trust in the central authority controlling the entire network. To facilitate data sharing in VSNs, this paper proposes a new scheme using consortium blockchain to realize secure data sharing. Nodes in the blockchain invoke smart contracts and implement the location-based Speculative Byzantine Fault Tolerance (LSBFT) to accomplish data-sharing transactions among vehicles. The scheme not only ensures the security of vehicle information but also protects the privacy of the shared data. Security analysis demonstrates that the proposed scheme can resist attacks and has shown transaction fairness, data confidentiality, non-repudiation, and traceability. Simulation results show that the scheme has higher sharing efficiency and less time to reach a consensus in the data storage process.**

***Index Terms*—Data sharing, Vehicular Social Networks (VSNs), consortium blockchain, smart contract, location-based Speculative Byzantine Fault Tolerance (LSBFT), transaction fairness.**

This work was supported in part by the National Natural Science Foundation of China under Grants 52331012 and 61672338, the National Key Research and Development Program of China under Grant 2021YFC2801001, the Natural Science Foundation of Shanghai under Grant 21ZR1426500, the Imperial-Nanyang Technological University Collaboration Fund under Grant INCF-2021-007, the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101006411, the Top-notch Innovative Talent Training Program for Graduate students of Shanghai Maritime University under Grant 2021YBR008, and the China Scholarship Council under Grant 202208310194. *(Corresponding author: Kuan-Ching Li, Dezhi Han)*.

Mingming Cui and Dezhi Han are with the College of Information Engineering, Shanghai Maritime University, Pudong 201306, China (e-mail: mmcui@stu.shmtu.edu.cn; dzhan@shmtu.edu.cn).

Han Liu is with the College of Transport & Communications, Shanghai Maritime University, Pudong 201306, China (e-mail: liuhanshmtu@163.com).

Kuan-Ching Li is with the Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan (e-mail: kuancli@pu.edu.tw).

Mingdong Tang is with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510420, China (e-mail: mdtang@gdufs.edu.cn).

Chin-Chen Chang is with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan (e-mail: alan3c@gmail.com).

Ferheen Ayaz and Zhengguo Sheng are with the Department of Engineering and Design, University of Sussex, Brighton BN1 9RH, U.K. (e-mail: f.ayaz@sussex.ac.uk; z.sheng@sussex.ac.uk).

Yong Liang Guan is with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: eylguan@ntu.edu.sg).

I. INTRODUCTION

IN recent years, the number of vehicles on the road has increased rapidly, putting significant pressure on the traffic situation in several areas worldwide. To improve road mobility and meet the needs of vehicle interaction, Vehicular Ad Hoc Networks (VANETs) have been proposed to provide a Peer-to-Peer (P2P) data transmission service for vehicles within a specific range [1]. With the rapid development of P2P communications, the influence of social behavior also gradually rises in many network systems, including VANETs. Vehicular Social Networks (VSNs) are developed by incorporating social behaviors of vehicular communications into VANETs [2]. VSNs manifest primarily in vehicles interacting with each other with the same destination or similar interests [3]. In VSNs, vehicles can exchange all types of information with each other through Roadside Units (RSUs), including traffic jams, road maintenance, accidents, weather warnings, and roadside construction [4]. By sharing information about roadside constructions, vehicles can find their destination more efficiently; by sharing road conditions, vehicles can avoid unnecessary traffic accidents and plan better travel routes.

Due to the wireless communication mode adopted by VSNs, communication among vehicles meets several security challenges that seriously threaten vehicle users' lives and property safety [5]-[8], such as information theft, tampering, forgery, and even malicious abuse. For example, driving information is illegally stolen. Then the driver's address, travel track, and other private information are leaked, and even out of interest, the stolen information is abused in different scenarios, which will endanger the personal safety of the driver, deliberately tamper with the information transmitted in the network about traffic accidents into road safety, or directly forge false road information, resulting in the vehicle receiving the wrong signal, and then the driver to engage in dangerous behavior. Concerning the security of the data transmission process, a straightforward approach is to encrypt the transmitted data via symmetric encryption and asymmetric encryption, i.e., vehicle users can use private keys of different pseudonyms to encrypt the transmitted data and maintain data confidentiality [9]. However, obtaining the decrypted data for the user requesting data sharing is a challenging problem. Currently, available methods to deal with this issue fall into two categories: 1) the data requester first gets the data owner's encrypted data, and then the data owner sends the decryption key to the requester, or 2) a third party decrypts the shared data encrypted by the data owner and then sends the data to the data requester. The computing overhead of users in the latter

approach is typically lower than in the former, despite the last may raise security issues with shared data.

The emerging blockchain technology offers new directions for solving VANETs' security problems [10]. As Bitcoin's underlying technology, blockchain consists of several data blocks linked chronologically, where transactions are recorded like a database ledger. One advantage is that its applications can run distributed, removing the central authority controlling the entire network. Blockchain technology refers to a computing paradigm where encrypted and chained blocks are used to store data, and a consensus mechanism is utilized to verify and update blocks. Smart contracts are invoked to manipulate the information in blocks [11], whilst blockchain complements the decentralization feature of VANETs to offer load balancing and avoid single-node failures. To ensure trust through consensus and security through a smart contract, building VSNs based on blockchain is reasonable.

Traditional public blockchain requires all nodes in the network to implement the consensus algorithm, resulting in the massive consumption of network resources. In contrast, the consortium blockchain uses Pre-Selected Nodes (PSNs) in VSNs to complete the consensus process and save network resources [12]. The consensus mechanism most commonly adopted by consortium blockchain is the Practical Byzantine Fault Tolerance (PBFT) algorithm, which achieves the state consistency of all nodes through mutual transmission and verification between nodes, reducing the complexity of traditional Byzantine protocols though only suitable for scenarios with a small number of consensus nodes. A smart contract is a program deployed on the blockchain that controls digital assets, including agreements between users, which are automatically executed by the computer system [13], so vehicle data owners can utilize smart contracts to regulate the data access process of other vehicles in VSNs. Once the trigger conditions in the contract drawn by the vehicle are met, smart contracts automatically issue corresponding data resources and perform corresponding operations. As such, blockchain-based smart contracts not only fully embody the advantages in computing costs and execution efficiency but also can prevent the interference of malicious behaviors.

A. Motivation

Due to the transparency and openness of blockchain, there will be privacy leakage if the uploaded data is directly stored in plaintext. Thereafter, the data collected by vehicles must be encrypted before being uploaded to blockchain nodes and stored in the ciphertext. And since using the PBFT algorithm in VSNs with large consensus nodes will lead to low consensus efficiency, the Speculative Byzantine Fault Tolerance (SBFT) [14] consensus algorithm is suggested here. The SBFT algorithm simplifies the process of pairwise interaction of consensus nodes to verify messages in PBFT, minishing the time cost of reaching an agreement. For VSNs with specific network environments, the existing blockchain-based data sharing methods cannot well guarantee the system's security and efficiency simultaneously. In general, how to design a secure and efficient data-sharing scheme in VSNs is an insistent demand.

B. Contributions

To address the issues of blockchain transparency, improve the security of data and users, and optimize the sharing efficiency, this work proposes a secure data-sharing scheme for consortium blockchain-enabled VSNs (DSCBV). Considering the communication distance between the vehicle and the RSU, the Location-based SBFT (LSBFT) consensus algorithm is adopted in our proposed scheme, which is obtained by optimizing the selection method of primary nodes based on the SBFT algorithm. The main contributions of this work are summarized as follows.

1) The shared data is encrypted and signed before uploading and stored in the blockchain in ciphertext, preventing the specific content of the shared data and the privacy of the vehicle users from being leaked. As the consortium blockchain nodes, RSUs undertake most of the data encryption and decryption operations, mitigating the burden of vehicle users. The LSBFT consensus algorithm is implemented by the pre-selected RSUs, in which the RSU nearest to the vehicle user uploading data or requesting sharing is directly designated as the primary node, significantly reducing the time to reach consensus.
2) Smart contracts are constructed in which the relevant information about the vehicle users and the shared data they upload are mainly written. When the vehicle users request sharing to trigger the conditions, the RSUs invoke smart contracts, and the corresponding operations are performed to complete the process of data sharing and payment safely and fairly between vehicle users.
3) The proposed scheme's resistance to possible attacks, including consensus algorithm attacks, tampering attacks, impersonation attacks, replay attacks, man-in-the-middle attacks, and DDoS attacks, are analyzed. In addition, it is shown that the proposed scheme can achieve transaction fairness, decentralization, data confidentiality, non-repudiation, and traceability.
4) Performance evaluation is conducted from the aspects of computing and communication costs of the vehicle users as well as the throughput and delay of the consensus algorithm. Experimental results demonstrate that this proposed scheme has higher sharing efficiency of vehicle users and less time to reach a consensus in the data storage process.

The remainder of this article is organized as follows. The related work is discussed in Section II. The system model, assumptions, explanations, threat model, and security requirements of the proposed scheme are formalized in Section III, and the proposed scheme is described in Section IV. The security and performance of the proposed scheme are analyzed in Sections V and VI, and finally, concluding remarks and future directions are drawn in Section VII.

II. RELATED WORK

A. Traditional Data Sharing Schemes

Several schemes have been proposed to obtain shared data

TABLE I
COMPARISON AMONG EXISTING SCHEMES

Schemes	[18]	[19]	[20]	[21]	[22]	[23]	[34]	[35]	[36]	[37]	[39]	[40]	[41]	DSCBV
VSNs	×	×	×	√	√	√	×	×	×	×	√	√	√	√
Decentralized Network	×	×	×	×	×	×	√	√	√	√	√	√	√	√
Non-Repudiation	–	–	×	×	–	–	–	×	–	–	×	√	–	√
Identity Authentication	–	–	×	√	–	–	–	×	×	–	×	√	–	√
Privacy protection	√	√	√	√	√	×	–	√	√	√	√	×	√	√
Low Sharing Efficiency	×	√	×	–	×	–	–	–	–	–	–	√	×	√

with requesting users [15]-[17]. Liang *et al.* [18] proposed a cloud data-sharing scheme by adopting Attribute-Based Encryption (ABE) and setting the ciphertext policy to achieve secure data sharing. However, in this scenario, the data owner must always be online to generate different decryption keys for other users. Wong *et al.* [19] proposed a secure Searchable Encryption (SE) scheme that encrypts data with item keys to achieve data interoperability. The Cloud Service Provider (CSP) assumes a part of the computation of data owners in the data query process, dramatically reducing their computational burden. This scheme, however, does not mention data sharing between users, aiming at querying encrypted data. Zhou and Wang [20] proposed a half-decrypted data-sharing scheme to protect data privacy, where data owners can share their data with different users without exposing their decryption keys to adversaries. However, this scheme must reflect shared user information, making it challenging to record shared information and track users. Moreover, the data that the user requests for access is determined by the data owner, resulting in the user being unable to obtain specific data according to particular requirements. As such, this scheme is unsuitable for data sharing between vehicles in VSNs because it does not fit the practical application.

Chen *et al.* [21] put forward a message forwarding scheme between vehicles, capable of verifying the message's integrity but failing to realize the vehicles' non-repudiation. Schlegel *et al*. [22] proposed a location-sharing scheme in VSNs, allowing users within a group to get shared location information. The user's location can be dynamically updated as it changes, and the maximum distance shared within the group can be set. Nevertheless, the server can only know the location of other users in different groups by comparing large distances. Xiao *et al*. [23] established a particular mapping protocol between the server and the base station to share the location information of vehicles. The vehicle user sends the base station a request for the location information of other vehicles, and then the base station returns the result to the user by querying the server. However, the server is fully aware of all the location information stored in this study, which may lead to the disclosure of sensitive data. Due to the limitations of large and complex networks on cloud servers, none of the above schemes is suitable for data sharing in resource-constrained networks[24].

B. Data Sharing Schemes Based on Blockchain

The development of blockchain has brought solutions to the abovementioned problems, as its decentralized distributed storage structure is suitable for data sharing in complex networks, and thus, increasingly more data-sharing models based on blockchain have been studied [25]-[31]. Sun *et al.* [32] proposed a blockchain-based shared service model to realize data sharing between users. However, this study only presents concepts and building models rather than implementing the solution. Yue *et al.* [33] proposed a data-sharing scheme based on blockchain to assist with patients' queries and share their medical data on the premise of protecting data privacy. This scheme uses a purpose access model to provide complete control of medical data to corresponding patients. Even though this study mentions the potential of secure multi-party computing, a concrete solution still needs to be implemented. Xia *et al.* [34] proposed a blockchain-based data-sharing scheme to facilitate users' access to shared data and preserve the users' privacy. Nevertheless, the authentication algorithm between entities is not defined in this scheme. Xia *et al.* [35] put forward a secure data-sharing framework based on blockchain in the cloud environment, which entirely resolves the access control issues of stored data but does not apply to the scenario of VANETs.

Recently, data-sharing schemes based on blockchain technology have been proposed specifically for VSNs [36]-[38]. Zhang and Chen [39] proposed a secure data-sharing model based on the consortium blockchain in VANETs. In this model, the bilinear pairing technique is utilized to guarantee data integrity. However, the authenticity of data can only be guaranteed by considering the additional identity authentication in the communication process. Kang *et al.* [40] proposed a reputation-based data-sharing scheme among vehicles by combining consortium blockchain and smart contracts. As blockchain nodes, RSUs are not fully trusted; therefore, the raw data obtained by the RSUs is not secure. Furthermore, data requesters not only need to download all data blocks to determine the data they want before sending the request but are also responsible for generating shared records, which reduces the sharing efficiency of data requesters; more significantly, there is a risk of unfairness as data requesters would only pay once they have access to the shared data. Fan *et al.* [41] proposed a data-sharing scheme in VSNs combined with Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and blockchain to achieve one-to-many information security sharing. However, data owners of the scheme are designated as members of the consortium blockchain, consisting of head members and ordinary members. The consensus algorithm is performed to select the head member to upload the information.

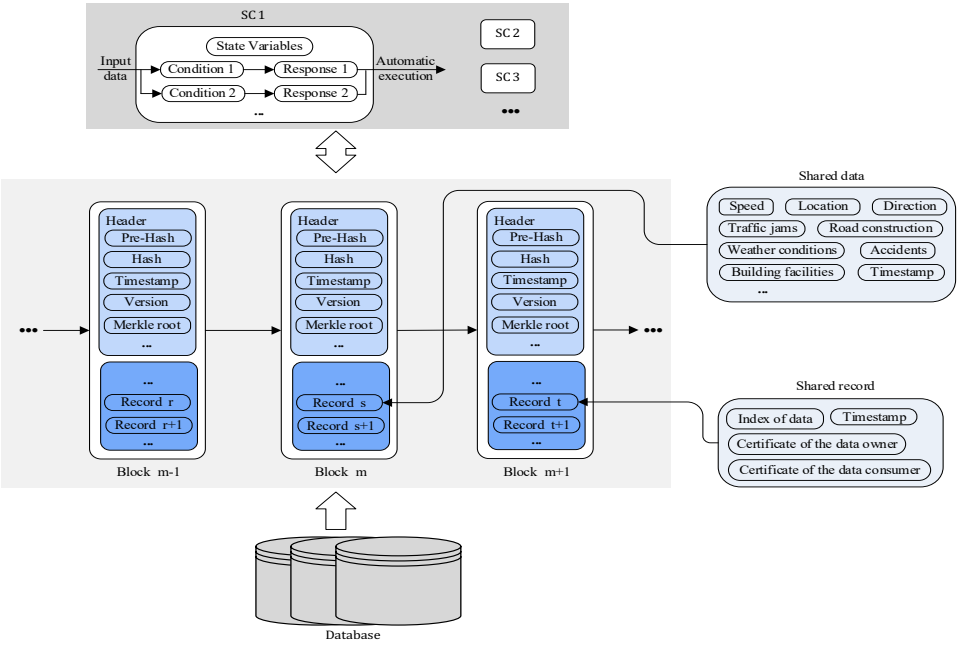

Fig. 2. Structure of VCB.

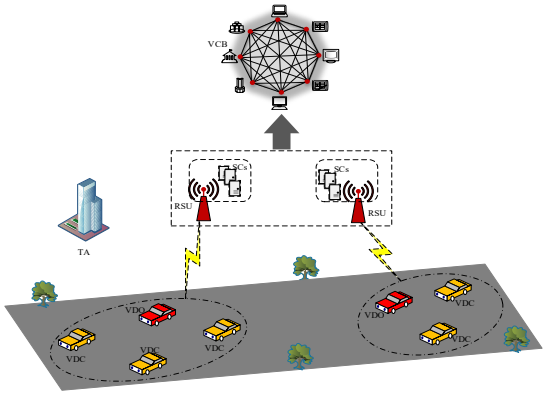

Fig. 1. System architecture.

Besides, the data owner must verify the correctness of the encrypted data uploaded to the CSP, reducing the sharing efficiency.

To clearly illustrate the difference between the related work and the proposed DSCBV scheme, an in-depth comparison is shown in Table I, where "√", "×", and "−" mean satisfied, dissatisfied, and uninvolved, respectively. Among all data-sharing schemes for VSNs, only our proposed scheme can realize non-repudiation, identity authentication, privacy protection, and low sharing efficiency simultaneously in a decentralized network.

III. PRELIMINARIES

This section introduces the proposed scheme's system model, assumptions, explanations, threat model, and security requirements.

A. System model

As shown in Fig. 1, the system model consists of Trusted Authority (TA), Vehicle Data Owners (VDOs), Vehicle Data Consumers (VDCs), RSUs, Smart Contracts (SCs), and Vehicle Consortium Blockchain (VCB). Among them, VDOs and VDCs are both vehicle users. All vehicles in the system are equipped with On-Board Units (OBUs) for data sensing in communication.

1) *TA*: The TA is a trusted third party with enough computing power to initialize the system and register VDOs, VDCs, and RSUs.
2) *VDOs*: In addition to their speeds, positions, and directions, VDOs also collect other data, such as road construction and weather conditions, among others. The sensed data is encrypted by the VDO and uploaded to the nearest RSU for sharing with VDCs. Before uploading the data, the VDO sends the formulated SC from the secure channel to the RSU for deployment. The VDO can get paid accordingly when other vehicles obtain the shared data it uploads.
3) *VDCs*: Whether a VDC wants information about a close or distant road, it sends a request to the nearest RSU for data sharing. After paying for the required data, the VDC can get the corresponding data. Also, the VDC can upload its data to the RSU for sharing, in which case it plays the role of the VDO.
4) *RSUs*: An RSU has enormous computing and storage power. As the nodes of the ledger for VCB, RSUs work through a consensus algorithm so that each RSU stores consistent content. The RSU can process and store data uploaded by the VDO and reply to the VDC after receiving the VDC's request for sharing.
5) *SCs*: Each RSU contains several SCs received from VDOs. A SC is a contract predefined by the VDO that automatically sends the corresponding response as long as

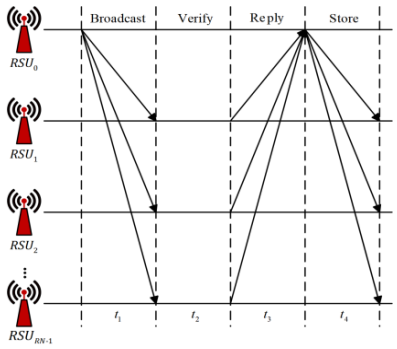

Fig. 3. Consensus process.

the specified conditions are met; once established, the contract cannot be modified. The proposed scheme's main contents include the VDO's certificate, the index of shared data, the description of shared data, the decryption key, the price, and the corresponding payment address. Data-sharing transactions between the VDO and the VDC are done by SC execution. After the VDC sends the request for sharing, it will search for the corresponding data in the block. Later, when the VDC pays for the shared data correctly, it will send the payment and the data to the corresponding VDO and VDC, respectively, and publish the shared record to the VCB through the detailed consensus algorithm described below. The SC saves transaction costs and introduces automatization, improving accuracy.

6) *VCB*: The VCB can be seen as a decentralized and reliable data platform linked together as a trading chain. It consists of a set of RSUs that store information about uploaded encrypted data and records of sharing events between vehicles. The structure of the VCB is shown in Fig. 2, including SCs, blocks, and databases. The block header mainly includes the previous block's hash, the current block's hash, timestamp, version number, Merkle root, etc. The block body includes data and payment transaction records stored by RSUs.

B. Assumptions and explanations

The following assumptions and illustrations are presented to make the proposed scheme complete and easier to understand.

1) It is assumed that RSUs are honest but curious. As nodes of the consortium blockchain, they are responsible for participating in the consensus algorithm to generate blocks. Nevertheless, they may leak stored data to malicious users for personal gain. Due to this factor, the RSUs are not entirely trustworthy.
2) In the VCB, the RSUs could become Byzantine nodes for various malicious attacks. Therefore, suppose $RN \geq 3f + 1$, where f is the number of Byzantine nodes and RN is the number of RSUs participating in the consensus process [42]. That is, most RSUs are honest nodes.
3) For the LSBFT consensus algorithm, each round of PSNs consists of a primary node and multiple secondary nodes, in which the primary node is specified as the RSU closest to the vehicle user who uploads data or requests sharing. At the same time, RSUs with better hardware and software environments are selected as the secondary nodes to participate in the consensus process [43]. As shown in Fig. 3, the detailed process includes broadcasting, verifying, replying, and storage. The RSU_0 is the primary node, while the other RSUs are the secondary nodes. Once the primary node receives the upload information during the consensus process, it will send the computed message to other PSNs, who validate it separately and broadcast the result to the primary node. Finally, the primary node RSU broadcasts the data block about the information to all other RSUs for storage as long as the feedback from all the secondary nodes is consistent.
4) Throughout the process, the VDO only communicates with the RSU closest to it, and the VDC only communicates with the RSU most relative to it.

C. Threat model

The proposed system model is vulnerable to internal and external attacks, which can interfere with the regular communication of vehicles. Based on the assumptions mentioned above that most nodes in the VCB are honest, a few malicious nodes fail to pass the consensus algorithm and thus cannot perform any operations on the stored information. Therefore, this work mainly considers attacks of external adversaries on VSNs combined with the consortium blockchain that serve the following attacks: consensus algorithm attacks, tampering attacks, impersonation attacks, replay attacks, man-in-the-middle attacks, and DDoS attacks.

1) *Consensus algorithm attack:* The consensus algorithm stores information about shared data and records. Notably, the adversary attacks the PSNs as malicious nodes to prevent proper storage of this information.
2) *Tampering attack:* In this attack, the adversary tampers with the data uploaded by vehicles in the block to cause traffic chaos and may also tamper with the shared records stored in the block to prevent accurate shared information from being obtained.
3) *Impersonation attack:* An adversary may forge or control the PSNs to generate the shared data stored in the VCB to cause traffic chaos. They may also control the PSNs to develop the shared records between vehicles stored in the VCB, causing wrong shared information to be generated.
4) *Replay attack:* Under this attack, an adversary tries intercepting and replaying messages transmitted in VSNs. This attack can trick the nodes in the VCB by repeatedly requesting shared data so that the adversary can benefit from the sharing process.
5) *Man-in-the-middle attack:* In the insecure transmission channel, the adversary intercepts the public keys issued by TA, replaces the public keys of vehicle users and RSUs with its public keys, and obtains or even modifies the real information transmitted by intercepting the encrypted information, to control the communication between vehicle users and RSUs.
6) *DDoS attack:* This attack degrades network performance by sending many requests to the VCB's nodes, paralyzing the system with bandwidth and transmitted power

TABLE II
NOTATIONS

Symbol	Definition
D_i	The ith block of data D
N_i	The index of the data block D_i
Dep_i	The description of the content of block D_i
Pr_i	The price payable for D_i
Pay_i	The payment for D_i from the VDC
Wd_{VDO}	The wallet address of the VDO
k_i	The encryption key for D_i based on AES
pk_x	The public key of entity x
sk_x	The private key of entity x
$cert_x$	The certificate of entity x
$En(pk_x, y)$	Encryption of information y based on ECC with the public key of entity x
$De(sk_x, y)$	Decryption of information y based on ECC with the private key of entity x
$En(k_i, y)$	Encryption of information y based on AES with the key k_i
$De(k_i, y)$	Decryption of information y based on AES with the key k_i
$Sign(sk_x, y)$	Signature of information y with the private key of entity x
$x\|\|y$	Element x concatenates to element y
$timestamp$	The time record of the current event

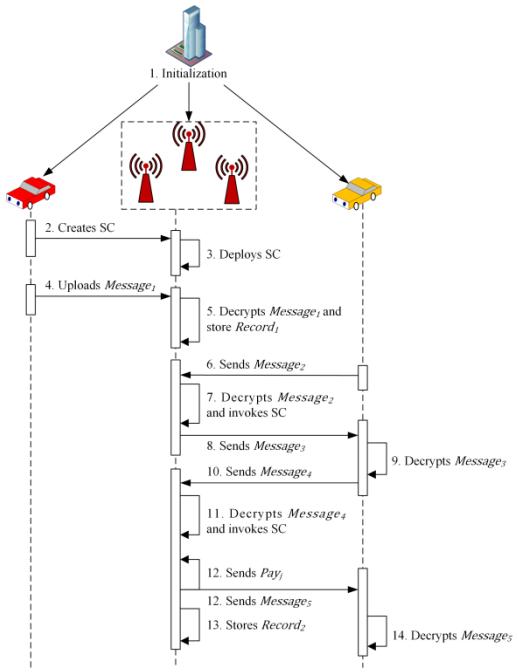

Fig. 4. Process of the DSCBV scheme.

limitations, thereby preventing the RSUs from providing shared access to vehicle users.

D. Security requirements

The following requirements should be considered when designing a secure data-sharing scheme.

1) *Transaction fairness:* The final shared data will be available once the VDC makes the correct payment for the requested data, and accordingly, the VDO will receive the due remuneration. Conversely, the VDC that does not pay or sends the wrong payment cannot get the shared data.
2) *Decentralization:* Safe and convenient data sharing can be carried out between vehicles without any trusted intermediary.
3) *Data confidentiality:* The adversary must not get any information in the transmission process, and the RSUs cannot get the real content of the encrypted data stored.
4) *Non-repudiation:* In communication, the vehicle user who uploads the shared data cannot deny his behaviors. Likewise, the vehicle user requesting the shared data cannot restrict his behaviors.
5) *Traceability:* After the sharing process is completed, the RSUs can trace back to the information of the VDO that uploaded the shared data and the input of the VDC that sent the request for sharing.

IV. PROPOSED SCHEME

The proposed DSCBV scheme comprises four phases: initialization, SC deployment, data uploading, and data sharing. The notations and their definitions used in the scheme are listed in Table II. Implementing this scheme, we assume that the VDO formulates the contract SC according to the data to be uploaded, and the VDC wants to share a portion of the data. The closest node to the VDO is denoted by RSU^{α}, while RSU^{β} indicates the nearest node to the VDC. Fig. 4 is an overview of the process of DSCBV, and additional details follow next.

A. Initialization phase

The TA executes the algorithm to register VDOs, VDCs, and RSUs. In this process, the TA uses Elliptic Curve Cryptography (ECC) for initialization. After verifying the identity of each entity, the TA generates public-private key pairs, certificates, and wallet addresses for them. The VDO retrieves its public key pk_{VDO}, private key sk_{VDO}, certificate $cert_{VDO}$, and wallet address Wd_{VDO}, while the VDC gets its public key pk_{VDC}, private key sk_{VDC}, and certificate $cert_{VDC}$. Similarly, the public key, private key, and certificate of the node RSU^{α} are $pk_{RSU^{\alpha}}$, $sk_{RSU^{\alpha}}$, and $cert_{RSU^{\alpha}}$, respectively. Meanwhile, the public key, private key, and certificate of the node RSU^{β} are $pk_{RSU^{\beta}}$, $sk_{RSU^{\beta}}$, and $cert_{RSU^{\beta}}$, respectively. The wallet address corresponds to the wallet account and is generated by random pseudonyms to protect the vehicle user's privacy. Throughout the communication, the certificate of each entity can be used to confirm the identity of the entity.

B. SC deployment phase

The VDO and the RSUs execute the algorithm. In this phase, the VDO adopts the Advanced Encryption Standard (AES) to generate keys for the uploaded data. The VDO enters the keys and data-related information into the SC, then sends it for the RSU^{α} to be deployed to each node of the VCB.

1) The VDO divides the shared data D to be uploaded into n blocks $\{D_1, \cdots, D_i, \cdots D_n\}$, and then generates keys $\{k_1, \cdots, k_i, \cdots k_n\}$ for each block to encrypt and decrypt the data blocks.
2) The VDO inputs $\{k_i||N_i||Dep_i||cert_{VDO}||Pr_i||Wd_{VDO}||timestamp\}_{i\in[1,n]}$ into the SC, where N_i is the index of

the data block D_i, Dep_i is a description of the real content of D_i, Pr_i is the price payable for D_i, and $timestamp$ refers to the time of the current event. N_i combines Dep_i to form an index table for a quick search of the data requested by the VDC, and $timestamp$ is embedded into the transmitted information can effectively avoid tampering and replay attack.

3) The VDO defines the SC's response conditions and corresponding response operations. After receiving the VDC's request, the SC searches and finds the data information in the block; after receiving the correct payment from the VDC, the SC sends the encrypted information and payment for the shared data to the VDO and the VDC, respectively. Then, the SC generates the shared record and publishes it to the VCB.
4) After receiving the SC sent by the VDO, the PSNs implement the LSBFT-based consensus algorithm. RSU$^{\alpha}$, as the primary node generates a new block for the SC and broadcasts it to the other secondary nodes. Then, the secondary nodes verify the block and return the verification results to the primary node. Finally, the primary node broadcasts the SC to all other RSUs for deployment as long as the feedback from all the secondary nodes is consistent. The main steps of the consensus process for the DSCBV scheme are described below.
   a) The primary node RSU_0 generates the information $Req\_consensus$ about the data and broadcasts it to the other secondary nodes in the VCB.
   $$Req\_consensus=(data||hash(data||timestamp)||cert_{RSU_0}||Sig_{RSU_0}||timestamp), \quad (1)$$
   where
   $$Sig_{RSU_0}=Sign(sk_{RSU_0},data||hash(data||timestamp)) \quad (2)$$
   b) The secondary node RSU_k verifies the received $Req\_consensus$ and returns the verification result $Result_k$ to the RSU_0.
   $$Result_k=En(pk_{RSU_0},res\_ver_k||cert_{RSU_k}||Sign(sk_{RSU_k},res\_ver_k)||timestamp) \quad (3)$$
   c) The RSU_0 first compares the received res_ver_k, then generates the new block $Block$ and broadcasts it to all nodes in the VCB for storage on the premise that the results are consistent.
   $$Block=(data||hash(data||timestamp)||cert_{RSU_k}||Sig_{RSU_k}||timestamp), \quad (4)$$
   where
   $$Sig_{RSU_k}=Sign(sk_{RSU_k},data||hash(data||timestamp)||cert_{RSU_k}) \quad (5)$$

C. Data uploading phase

The VDO and the RSUs are involved in this phase. The VDO uses AES and Digital Signature Algorithm (DSA) to encrypt and sign the uploaded data and sends the encrypted data and related information to the RSU$^{\alpha}$. Then, the RSU$^{\alpha}$ processes the uploaded data, and all the RSUs generate new blocks to link to the VCB through the LSBFT-based consensus algorithm.

1) The VDO encrypts the data block D_i with the key k_i using AES, represented as $En(k_i,D_i)$, and sign it using DSA, represented as $Sign\left(sk_{D_i},En(k_i,D_i)\right)$, where sk_{D_i} and pk_{D_i} are the private-public key pair of D_i used for the signature and signature's verification of the encrypted data block.
2) The VDO generates the information $Message_1$ for the uploaded data and sends it to the RSU$^{\alpha}$.
   $$Message_1=En(pk_{RSU^{\alpha}},\{En(k_i,D_i)||N_i||Sig_i||cert_{VDO}||timestamp\}_{i\in[1,n]}), \quad (6)$$
   where
   $$Sig_i=Sign\left(sk_{D_i},En(k_i,D_i)\right) \quad (7)$$
3) Once $Message_1$ is received, the RSU$^{\alpha}$ decrypts $Message_1$ with $sk_{RSU^{\alpha}}$ and validates the correctness of the encrypted data through Sig_i. If it is not correct, the RSU$^{\alpha}$ refuses to store the uploaded data; otherwise, the LSBFT-based consensus algorithm is executed to ensure that all RSUs in the VCB store the block with the content $Record_1$.
   $$Record_1=\{En(k_i,D_i)||N_i||cert_{VDO}||timestamp\}_{i\in[1,n]} \quad (8)$$

D. Data sharing phase

The VDO, the VDC, the RSUs, and the SC perform the algorithm. During this processing, the VDC sends a shared request to the RSU$^{\beta}$, and the RSU$^{\beta}$ calls the SC to search for the encrypted data information that the VDC wants and sends it to the VDC. Next, the VDC sends the payment to the RSU$^{\beta}$ to retrieve the requested data, while the VDO receives the payment through the RSU$^{\alpha}$. Finally, the SC generates the shared record and publishes it to the VCB. Algorithm 1 shows all the operations that are done by invoking the SC.

1) The VDC generates the requested information $Message_2$ and sends it to the RSU$^{\beta}$, where Req includes a description of the data block D_j.
   $$Message_2=En(pk_{RSU^{\beta}},Req||cert_{VDC}||pk_{VDC}||timestamp) \quad (9)$$
2) On receipt of $Message_2$, the RSU$^{\beta}$ decrypts it with $sk_{RSU^{\beta}}$, then invokes the SC to search the index table according to Req, and finds the information of D_j through the index N_j. Ultimately, the RSU$^{\beta}$ sends the requested data information $Message_3$ to the VDC.
   $$Message_3=En(pk_{VDC},cert_{RSU^{\beta}}||N_j||Pr_j||timestamp) \quad (10)$$
3) After receiving $Message_3$, the VDC first decrypt it with sk_{VDC} to obtain Pr_j, then generates $Message_4$ and sends $Message_4$ to the RSU$^{\beta}$, where Pay_j is the payment for D_j with the corresponding price Pr_j.
   $$Message_4=En(pk_{RSU^{\beta}},N_j||Pay_j||cert_{VDC}||timestamp) \quad (11)$$
4) Once $Message_4$ is received, the RSU$^{\beta}$ decrypts it with $sk_{RSU^{\beta}}$ to get Pay_j. If Pay_j doesn't match Pr_j, the transaction will fail. Otherwise, the SC is immediately

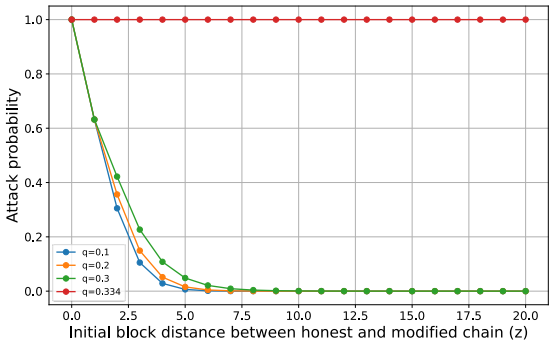

Fig. 5. Probability of tamper attack.

invoked to decrypt $En(k_i, D_i)$ with k_j to get D_j and generate $Message_5$. Then, the RSU$^{\beta}$ returns $Message_5$ to the VDC and the SC uploads Pay_j to the VDO's wallet address Wd_{VDO}, respectively. Finally, the LSBFT-based consensus algorithm is executed to ensure that all RSUs in the VCB store the block with the shared record $Record_2$, which is generated by the SC.

$$Message_5 = En(pk_{VDC}, D_j||N_j||cert_{VDO}||cert_{RSU^{\beta}}||timestamp) \quad (12)$$

$$Record_2 = cert_{VDO}||cert_{VDC}||N_j||timestamp \quad (13)$$

5) After receiving $Message_5$, the VDC decrypts it with sk_{VDC} to get D_j.

V. SECURITY ANALYSIS

In this section, we analyze the security of the proposed DSCBV scheme, including the resistance against attacks and security requirements.

A. Resistance against attacks

In this subsection, we explain how the proposed scheme can resist the attacks mentioned in Section III. C above. Among them, the first four types of resistance are mainly aimed at blockchain-related security attacks, including resistance to attacks of consensus algorithms, and on-chain content tampering, impersonation, and replay.

1) *Resistance against consensus algorithm attacks:* Assume that QN is the minimum number of nodes required to determine a consistent result in the consensus process. a) When all Byzantine nodes actively destroy the consensus process, the consensus algorithm can be implemented only if $RN - f \geq QN$; b) If Byzantine nodes make the different results sent between nodes pass, the consistency of the system cannot be realized. Therefore, honest nodes must exist in the intersection of any two node sets with different results, that is, $2QN - RN > f$ must be guaranteed. From the above two inequalities, $2RN - 2f \geq 2QN > RN + f$ is obtained and then $RN > 3f$ can be derived, that is, $RN \geq 3f + 1$. Only when the adversary has at least 1/3 or more computing power can the consensus algorithm be broken and the entire data of the VCB be controlled. Therefore, on the premise that $RN \geq 3f + 1$ is guaranteed, the proposed scheme can defend against the attack of the consensus algorithm.

2) *Resistance against tampering attacks:* For the distributed consortium blockchain, an adversary can only attack a part of the nodes to make them malicious nodes to tamper with the stored information, the shared data $Record_1$ and the shared record $Record_2$. However, when the consensus algorithm is executed, the tampered information does not pass the validation and thus cannot be written into the VCB's blocks by all nodes. More specifically, a simulation experiment is performed on Matlab to prove that the adversary could not tamper with the data in the VCB. Assuming that the block generation time in the blockchain is uniform, the length increment of the chain follows the Poisson distribution. The probability of a malicious node tampering attack is shown in Fig. 5, where q is the probability of the tampering chain competing for the next block. IIf the block distance of the honest chain and the modified chain is the same, only when the malicious node has mastered more than 1/3 of the computing power can it control all the data of the blockchain. So, our proposed scheme can defend against the tampering attacks on the premise that $RN \geq 3f + 1$ is guaranteed.

3) *Resistance against impersonation attacks:* According to the uploaded $Message_1 = En(pk_{RSU^{\alpha}}, \{En(k_i, D_i)||N_i||Sig_i||cert_{VDO}||timestamp\}_{i\in[1,n]})$, the encrypted shared data $En(k_i, D_i)$ is bound to $Sig_i = Sign(sk_{D_i}, En(k_i, D_i))$, so the adversary cannot successfully forge the signature Sig_i of encrypted shared data without knowing the private key sk_{D_i} of the signature. When the adversary uploads the forged information of shared data, the node RSU will refuse to store the wrong data information after verifying that the signature is incorrect. Moreover, in the distributed consortium blockchain, the adversary can only attack a part of the nodes, making them malicious nodes to forge $Record_1$ and $Record_2$. Still, when the consensus algorithm is executed, the forged information does not pass the validation and thus cannot be written to the VCB's blocks by all nodes. No adversary can forge the data information in the blocks, so the proposed scheme is safe against an impersonation attack.

4) *Resistance against replay attacks:* The messages SC, $Message_1$, $Message_2$, $Message_3$, $Message_4$, and $Message_5$ may be obtained by an adversary through an insecure channel for reuse in the data sharing process. However, these transmitted messages all contain a timestamp verified by the receiver for random numbers and freshness. Since the adversary cannot know the random numbers, the RSUs will find that the message is not recent or has been modified and reject the message when a mismatched timestamp is sent. The replay attack is embodied by an experiment based on Hyperledger Fabric V1.4.1 uploading duplicate data to the blockchain. For example, the VCB has stored $Record_2$. Fig. 6 shows

Fig. 6. Record2 query result.

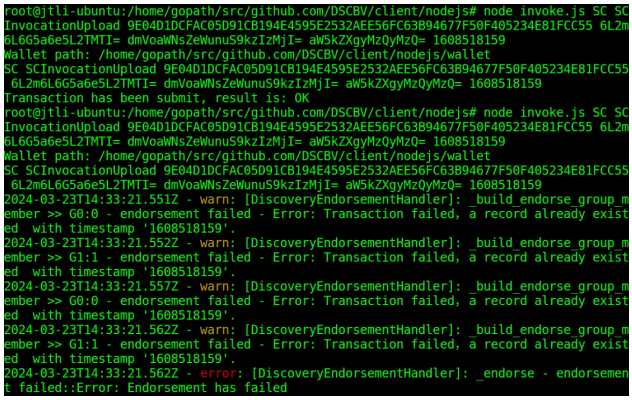

Fig. 7. Result of repeatedly uploading Record2.

the result of querying $Record_2$. The malicious node attempts to repeatedly upload $Record_2$ to the VCB for storage. Fig. 7 shows the result of repeatedly uploading $Record_2$ with each transaction failing and returning that the record already exists. Therefore, our proposed scheme can resist the replay attacks.

5) *Resistance against man-in-the-middle attacks:* In the initialization process, TA generates public keys for the VDO, the VDC, the RSU$^{\alpha}$, and the RSU$^{\beta}$, and generates corresponding certificates, namely $cert_{VDO}$, $cert_{VDC}$, $cert_{RSU^{\alpha}}$, and $cert_{RSU^{\beta}}$, respectively. The certificate contains the entity's identity and the public key's signature, which can be used to verify that the received public key corresponds to the signature. The adversary could not forge the certificates of vehicle users and RSUs, so the public key could not be replaced successfully. The proposed scheme enables vehicle users and RSUs to obtain the correct public key. Further, it guarantees the authenticity of the information transmitted in the communication process, preventing the man-in-the-middle attack.
6) *Resistance against DDoS attacks:* There are multiple dependent nodes in this distributed system. The entire system will not be affected even if any node goes down. Hence, the proposed scheme can effectively resist the DDoS attack.

B. Security requirements

This subsection demonstrates that the proposed DSCBV scheme meets the security requirements.

1) *Transaction fairness*: Only after receiving the correct payment from the VDC that matches the price Pr_j of the data block D_j, the RSU$^{\beta}$ will call the SC to send Pay_j and $Message_5 = En(pk_{VDC}, D_j||N_j||cert_{VDO}||cert_{RSU^{\beta}}||timestamp)$ to the VDO's account Wd_{VDO} and the VDC, respectively. In this case, the VDO will get Pay_j, and correspondingly, the VDC will get the requested data D_j by decrypting $Message_5$. If the VDC does not send a payment or sesends payment that does not match Pr_j, the RSU$^{\beta}$ wouldn't call the SC to send $Message_5$, resulting in the VDC not getting D_j.
2) *Decentralization*: Based on the decentralized storage mode of the consortium blockchain, the DSCBV scheme does not rely on any trusted third-party database to store data, avoiding the malicious attacks brought by traditional central data storage. The consensus mechanism in the blockchain ensures that each node of the VCB holds the same information record. Besides, the VDO and the VDC only need to interact with the nearest node, which makes it easy for the VDC to obtain the shared data uploaded by the VDO. By formulating and calling the SC, the transmitted data and corresponding reward can be obtained separately by the VDC and the VDO without going through a third-party agency.
3) *Data confidentiality*: During the upload data process, $Message_1$ uploaded by the VDO is generated after the encryption with the public key $pk_{RSU^{\alpha}}$ of the RSU$^{\alpha}$, so the adversary would not be able to find out the true content of $Message_1$ without knowing the private key $sk_{RSU^{\alpha}}$ of the RSU$^{\alpha}$. During the sharing process, $Message_2$ and $Message_4$ sent by the VDC are both generated after the encryption with $pk_{RSU^{\beta}}$, so the adversary would not be able to find out the true contents of $Message_2$ and $Message_4$ without knowing $sk_{RSU^{\beta}}$. Similarly, $Message_3$ and $Message_5$ sent by the RSU$^{\beta}$ are both generated after the encryption with the public key pk_{VDC} of the VDC, making it impossible for the adversary to know their true contents without knowing the private key sk_{VDC} of the VDC. In addition, the key k_i put into the SC cannot be obtained by the RSUs, resulting in the RSU being unable to decrypt $En(k_i, D_i)$ to get D_i.
4) *Non-repudiation*: After decrypting $Message_1 = En(pk_{RSU^{\alpha}}, \{En(k_i, D_i)||N_i||Sig_i||cert_{VDO}||timestamp\}_{i\in[1,n]})$ uploaded by the VDO, the RSU$^{\alpha}$ extracts $cert_{VDO}$ to confirm the VDO's identity information so that the VDO could not deny its operation of uploading $Message_1$; after decrypting $Message_2 = En(pk_{RSU^{\beta}}, Req||cert_{VDC}||pk_{VDC}||timestamp)$ and $Message_4 = En(pk_{RSU^{\beta}}, N_j||Pay_j||cert_{VDC}||timestamp)$ sent by the VDC, the RSU$^{\beta}$ extracts $cert_{VDC}$ to confirm the VDC's identity information so that the VDC cannot deny its operation of sending $Message_2$ and $Message_4$.
5) *Traceability*: At the end of the sharing process, the SC generates the shared record $Record_2 = cert_{VDO}||cert_{VDC}||N_j||timestamp$ and publishes it to the VCB. Through $Record_2$, the RSU can get the identity information $cert_{VDO}$ and $cert_{VDC}$ of vehicle users participating in the sharing process, which can be traced back to the VDO that uploaded the data and the VDC that sent the sharing request.

TABLE III
EXPERIMENTAL ENVIRONMENT

	Description
CPU	Intel(R) Core (TM) i7-8700 @3.20GHz
Memory	8GB
Operating System	Window 10
Programming Language	JAVA

TABLE IV
EXPERIMENTAL SETTINGS

Size of Each Data Block	Size of Uploaded Data	Number of Shared Requests
20MB	2GB	10
	4GB	20
	6GB	30
	8GB	40
	10GB	50
	12GB	60
	14GB	70
	16GB	80
	18GB	90
	20GB	100
	22GB	110
	24GB	120

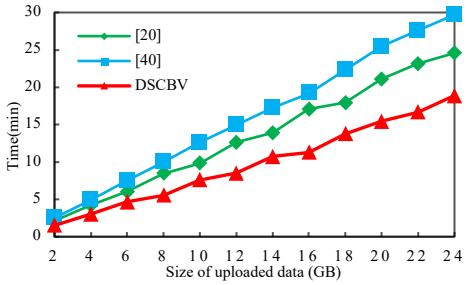

Fig. 8. Computational time of the VDO.

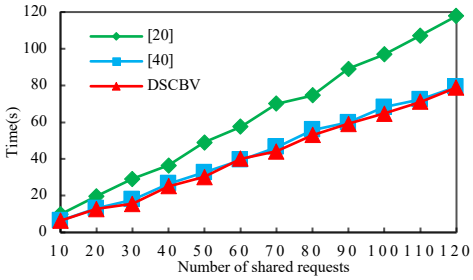

Fig. 9. Computational time of the VDCs.

VI. PERFORMANCE EVALUATION

In this section, we analyze the performance of the proposed DSCBV scheme from the aspects of computational cost and communication cost.

A. Computational cost analysis

A data sharing scheme for VSNs is expected to reduce the calculation cost of shared users on sharing; due to such, the performance evaluation is mainly focused on the calculation cost of the VDO and the VDC. Furthermore, the performance of the proposed DSCBV scheme is compared with [20] and [40]. Table III presents the environment where the experiment is conducted, while Table IV shows the settings of some experimental elements. Consider a scenario where only one vehicle uploads the shared data to the nearest RSU and more than one vehicle requests to share. In the following, some further experimental hypotheses and settings are presented.

1) Suppose that 2GB to 24GB of shared data are uploaded by the VDO over some time. We increase the amount of data per 2GB to test the total computing time of the VDO throughout the process. The size of each data block is supposed to be 20MB. Thus, 2GB data consists of about 100 blocks, and 24GB data consist of approximately 1200 blocks.
2) During this time, VDCs send requests for the shared data uploaded by the VDO. Specifically, a share request sent by the VDC corresponds to a shared data block.
3) For the uploaded 2GB to 24GB shared data, the VDCs send 10 to 120 requests for communicating with an increase in the amount of data per 10 to obtain the data. With a different number of share requests sent, the total computing time of the VDCs is tested during the entire sharing process.
4) For the scheme in [20], the VDO refers to the data owner, while the VDC refers to the shared user.

The calculation time of the VDO during the entire process is shown in Fig. 8. In these three schemes, the calculation time of the VDO increases almost linearly as the size of the uploaded data increases. When uploading 2GB to 24GB shared data, the computational cost of the VDO participating in the entire process ranges from 2.147min to 24.570min in scheme [20], from 2.528min to 29.683min in scheme [40], and in the proposed DSCBV scheme, the computational cost of the VDO ranges from 1.520min to 18.888min.

Considering the operations of the VDO in the abovementioned three schemes, the primary source of the VDO's computational overhead is analyzed. In scheme [20], in addition to generating the encrypted data, the VDO needs to send the keys for decrypting the encrypted data to the VDC and the TA. In scheme [40], in addition to uploading the encrypted data, the VDO also needs to receive sharing requests from the VDCs and send request information to the RUS. However, in DSCBV, the VDO only uploads the encrypted data, and all VDCs requesting sharing interact solely with the RSUs during the sharing process. Therefore, the calculation time of VDO in DSCBV is less than the other schemes.

Fig. 9 shows the calculation time of VDCs throughout the sharing process in the case of sending 10 to 120 share requests for 2GB to 24GB shared data. From the comparative schemes, the total computing time of VDCs increases linearly as the number of shared requests increases. The computational overhead of VDCs participating in the process ranges from 9.826s to 117.767s for scheme [20]; and from 6.387s to 79.210s for scheme [40], while for the proposed DSCBV scheme, the computational overhead of VDCs ranges from 6.319s to 78.862s.

Considering the operations of the VDC in the schemes, the primary source of the VDC's computational overhead is analyzed here. For scheme [20], the VDC needs to send a sharing request to obtain the encrypted data and then use its

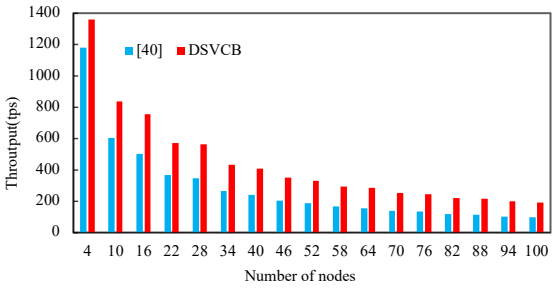

Fig. 10. Throughput of the consensus process.

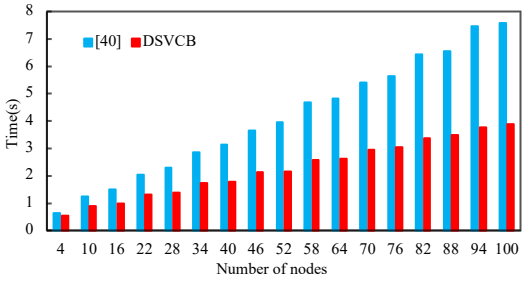

Fig. 11. Time cost of the consensus process.

private key and the key sent by the VDO successively to decrypt the encrypted data, to get the data shared by the VDO. For scheme [40], the VDC needs to send the request for sharing to obtain the encrypted data, then decrypt the data with its private key to get the shared data, and finally, send payment and generate the shared record. For the proposed DSCBV scheme, the VDC sends a share request, receives the encrypted data after payment, and decrypts it with its private key to obtain the shared data. In comparison, the VDCs' calculation time of scheme [40] and DSCBV scheme is less than that of [20], and even the time cost of these two schemes is similar. However, in scheme [40], the RSU can obtain the real content of the shared data in the process of decryption, and the VDC may not pay after getting the shared data, which is unfair and has security risks.

To further highlight the proposed DSCBV scheme's performance, the consensus processes of scheme [40] and the DSCBV scheme suitable for data sharing in VSNs are simulated. In the case of 750 transactions, the number of nodes participating in the consensus process is set as 4, 10, 16, 22, 28, 34, 40, 46, 52, 58, 64, 70, 76, 82, 88, 94, and 100 respectively to calculate the throughput and consensus delay. As seen in Fig. 10, the throughput of the two schemes decreases as the number of nodes increases. The throughput of the proposed DSCBV scheme ranges from 1361.162tps to 193.249tps, which is higher than that of the scheme [40], whose throughput goes from 1179.245tps to 99.023tps. As shown in Fig. 11, the consensus delay of the two schemes increases gradually with the increasing number of nodes. The time for reaching consensus in the scheme [40] ranges from 0.636s to 7.574s, while in the proposed DSCBV scheme, it ranges from 0.551s to 3.881s. The primary source of time overhead is analyzed regarding the operations that nodes reach consensus on in these two schemes. Compared with the scheme [40], in DSCBV, the direct designation of the primary node closest to the vehicle user

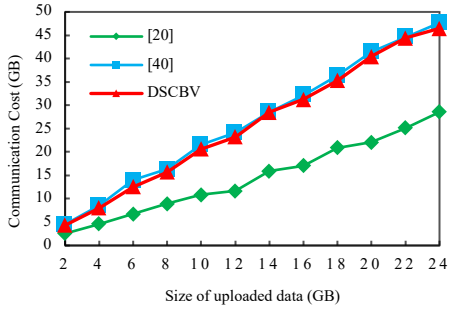

Fig. 12. Total communication cost.

uploading data or requesting sharing avoids the use of complex calculations. And the secondary nodes that receive the message from the primary node do not need to send the verification results to each other, instead, they return the verification results to the primary node. Therefore, the DSCBV scheme has less time to reach a consensus than others.

B. Communication cost analysis

Given the experimental assumptions and settings in the preceding subsection, the total communication overhead of VDO and VDCs participating in the whole process is calculated. Fig. 12 reveals the total communication cost of uploading 2GB to 24GB of shared data from VDO and sending corresponding 10 to 120 share requests from VDCs, growing as the volume of shared data increases. The total communication cost of the VDO and the VDCs ranges from 4.498GB to 48.889GB in the scheme [40], from 4.32GB to 48.112GB in the proposed DSCBV scheme, and from 2.547GB to 28.526GB in the scheme [20]. Next, the total communication cost is analyzed in the case of VDO uploading a data block D_i and VDC requesting the block.

Let $|D_i|$, $|k_i|$, $|N_i|$, $|Dep_i|$, $|cert_{VDO}|$, $|Pr_i|$, $|Wd_{VDO}|$, $|Req|$, $|cert_{VDC}|$, $|pk_{VDC}|$, and $|Pay_i|$ represent the size of D_i, k_i, N_i, Dep_i, $cert_{VDO}$, Pr_i, Wd_{VDO}, Req, $cert_{VDC}$, pk_{VDC}, and Pay_i, respectively. In the proposed DSCBV scheme, the VDO simply uploads the SC and $Message_1$ to the RSU$^{\alpha}$, so the main communication overhead of the VDO is $2|D_i|+|k_i|+2|N_i|+|Dep_i|+ 2|cert_{VDO}|+Pr_i|+|Wd_{VDO}|$. All the VDC needs to do is send $Message_2$ and $Message_4$ to the RSU$^{\beta}$, so the communication overhead of the VDC is $|Req|+2|cert_{VDC}|+|pk_{VDC}|+|N_i|+|Pay_i|$. We require $|N_i|$, $|Dep_i|$, $|cert_{VDO}|$, $Pr_i|$, $|Wd_{VDO}|$, $|Req|$, $|cert_{VDC}|$, $|pk_{VDC}|$, $|Pay_i|<<|k_i|<<|D_i|$, so the total communication overhead of the VDO and the VDC is about $2|D_i|$ =40MB. For scheme [40], the VDO must upload the encrypted data to the RSU and send the decryption key to the RSU upon receipt of the VDC's request. While the VDC needs to first apply to the VDO for sharing and then send the payment and the shared record to the VDO and the RSU, respectively, after getting the shared data. So the total communication overhead for the VDO and the VDC is about $2|D_i|$ =40MB. Similarly, for scheme [20], the VDO is required to upload the twice encrypted data to the CSP and send the decrypted keys to the TA and the VDC, respectively, while the VDC only needs to send the request for

sharing to the CSP, so the main total communication overhead of the VDO and the VDC is about $|D_i| = 20$MB.

The total communication overhead of the VDO and the VDC in the proposed DSCBV scheme is almost the same as that in scheme [40], which is more than $|D_i|$ in scheme [20]. Because the first two schemes upload the signature of encrypted data when uploading the encrypted data. In contrast, scheme [20] only uploads the encrypted data, resulting in the CSP's failure to verify the encrypted data uploaded by the VDO.

Overall, from the experimental results analyzed and observations, our proposed scheme performs better than schemes [20] and [40].

VII. CONCLUSIONS AND FUTURE WORK

This work proposes a scheme named DSCBV to realize secure data sharing in VSNs using the consortium blockchain technology. Smart contracts and a LSBFT-based consensus algorithm are utilized to ensure security and improve the efficiency of data sharing between vehicle users. The security analysis shows that the scheme can resist attacks and achieve transaction fairness, decentralization, data confidentiality, non-repudiation, and traceability. In addition, the performance analysis shows that the proposed scheme significantly outperforms the other two methods regarding the time consumption for reaching consensus and the computing cost for vehicle users. In future directions, a data selection method will be developed for vehicle data consumers to select more appropriate data among many similar ones uploaded by vehicle data owners. And the geospatial distribution and communication range of vehicles and RSUs will be considered, leading to a system model that is more applicable to real-world scenarios.

REFERENCES

[1] Y. F. Payalan and M. A. Guvensan, "Towards Next-Generation Vehicles Featuring the Vehicle Intelligence," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 30-47, 2020.

[2] S. Chavhan, D. Gupta, C. Nagaraju, R. A, A. Khanna, and J. J. P. C. Rodrigues, "An Efficient Context-Aware Vehicle Incidents Route Service Management for Intelligent Transport System," *IEEE Syst. J.*, vol. 16, no. 1, pp. 487-498, 2022.

[3] M. Cui, D. Han, J. Wang, K. -C. Li, and C. -C. Chang, "ARFV: An Efficient Shared Data Auditing Scheme Supporting Revocation for Fog-Assisted Vehicular Ad-Hoc Networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15815-15827, 2020.

[4] L. He, G. Ma, W. Qi, and X. Wang, "Charging an electric vehicle-sharing fleet," *Manuf. Service Oper. Manage.*, vol. 23, no. 2, pp. 471–487, 2021.

[5] J. Zhou, D. Tian, Y. Wang, Z. Sheng, X. Duan, and V. C. M. Leung, "Reliability-Optimal Cooperative Communication and Computing in Connected Vehicle Systems," *IEEE Trans. Mobile Comput.*, vol. 19, no. 5, pp. 1216-1232, 2020.

[6] M. Cui, D. Han, and J. Wang, "An Efficient and Safe Road Condition Monitoring Authentication Scheme Based on Fog Computing," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9076-9084, 2019.

[7] B. Palaniswamy, S. Camtepe, E. Foo, and J. Pieprzyk, "An Efficient Authentication Scheme for Intra-Vehicular Controller Area Network," *IEEE Trans. Inf. Forensics Security,* vol. 15, pp. 3107-3122, 2020.

[8] T. Limbasiya and D. Das, "Lightweight Secure Message Broadcasting Protocol for Vehicle-to-Vehicle Communication," *IEEE Syst. J.*, vol. 14, no. 1, pp. 520-529, 2020.

[9] D. Han, N. Pan, and K.-C. Li, "A Traceable and Revocable Ciphertext-policy Attribute-based Encryption Scheme Based on Privacy Protection," *IEEE Trans. Dependable Secure Comput.*, vol.19, no. 1, pp. 316-327, 2022.

[10] 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 Trans. Veh. Technol.*, vol. 69, no. 7, pp. 6856-6868, 2020.

[11] H. Li, D. Han, and M. Tang, "A Privacy-Preserving Storage Scheme for Logistics Data with Assistance of Blockchain," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4704-4720, 2022.

[12] D. Chattaraj, B. Bera, A. K. Das, S. Saha, P. Lorenz, and Y. Park, "Block-CLAP: Blockchain-Assisted Certificateless Key Agreement Protocol for Internet of Vehicles in Smart Transportation," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8092-8107, 2021.

[13] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.(CCS)*, vol. 16, pp. 254-269, 2016.

[14] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative Byzantine Fault Tolerance," *ACM SIGOPS Operating Systems Review*, vol. 27, no. 4, pp. 1-39, 2009.

[15] G. Li, R.Togo, T. Ogawa, and M. Haseyama, "Compressed gastric image generation based on soft-label dataset distillation for medical data sharing," *Computer Methods and Programs in Biomedicine*, 2022, vol.227, doi: 10.1016/j.cmpb.2022.107189.

[16] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing," *IEEE Trans. Depend. Secure Comput.*, vol. 16, no. 6, pp. 996-1010, 2019.

[17] C.P. Li, X.X. Li, P. Liu, W.J. Qiu, C.J. Yao, and B. Yuan, "Efficient and traceable data sharing for the Internet of Things in smart cities," *Computers and Electrical Engineering*, 2022, vol. 103, doi: 10.1016/j.compeleceng.2022.108389.

[18] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, Y. Yu, and A. Yang, "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Generation. Comput. Syst.*, vol. 52, pp. 95-108, 2015.

[19] W. K. Wong, B. Kao, D. W. L. Cheung, L. R, and S. M. Yiu, "Secure query processing with data interoperability in a cloud database environment," *Proc. IEEE ACM SIGMOD international conference on Management of data*, pp. 1395–1406, 2014.

[20] Y. Zhou and L. Wang, "SDS2: Secure Data-Sharing Scheme for Crowd Owners in Public Cloud Service," *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 22-29, 2017.

[21] C. Chen, M. Chiang, C. Peng, C. Chang, and Q. Sui, "A secure mutual authentication scheme with non-repudiation for vehicular ad hoc networks," *Int. J. Commun. Syst.*, vol. 30, no. 6, pp. e3081, 2017.

[22] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, "Privacy-preserving location sharing services for social networks," *IEEE Trans. Services Comput*., vol. 10, no. 5, pp. 811-825, 2017.

[23] X. Xiao, C. Chen, A. K. Sangaiah, G. Hu, R. Ye, and Y. Jiang, "CenLocShare: A centralized privacy-preserving location-sharing system for mobile online social networks," *Future Gener. Comput. Syst.*, vol. 86, pp. 863-872, 2018.

[24] S. Roy, S. Nandi, R. Maheshwari, S. Shetty, A. K. Das, and P. Lorenz, "Blockchain-Based Efficient Access Control with Handover Policy in IoV-Enabled Intelligent Transportation System," *IEEE Trans. Veh. Technol.,* 2023, doi: 10.1109/TVT.2023.3322637.

[25] Q. Kong, L. Su, and M. Ma, "Achieving Privacy-Preserving and Verifiable Data Sharing in Vehicular Fog With Blockchain," *IEEE Trans. Intell. Transp. Syst.*, vol.22, no.8, pp.4889-4898, 2021.

[26] K. O. -B. O. Agyekum, Q. Xia, E. B. Sifah, C. N. A. Cobblah, H. Xia, and J. Gao, "A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of Things Based on Blockchain," *IEEE Syst. J.*, vol. 16, no. 1, pp. 1685-1696, 2022.

[27] El-hacen Diallo, Omar Dib, and K.A. Agha, "A scalable blockchain-based scheme for traffic-related data sharing in VANETs," *Blockchain: Research and Applications*, vol. 3, no.3, pp. 100087, 2022.

[28] Y. Jiang, X. Shen, and S. Zheng, "An Effective Data Sharing Scheme Based on Blockchain in Vehicular Social Networks," *Electronics*, vol.10, no. 2, pp. 114, 2021.

[29] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-Health systems via consortium blockchain," *J. Med. Syst.*, vol. 42, no. 8, pp. 140, 2018.

[30] M. Firdaus and K.-H. Rhee, "On Blockchain-Enhanced Secure Data Storage and Sharing in Vehicular Edge Computing Networks," *Appl. Sci.*, vol. 11, no. 1, 2021.

[31] S. K. Dwivedi, R. Amin, S. Vollala, and A. K. Das, "Design of Blockchain and ECC-Based Robust and Efficient Batch Authentication Protocol for Vehicular Ad-Hoc Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 275-288, 2024.

[32] J. Sun, J. Yan, and K. Z. K. Zhang, "Blockchain-based sharing services: What blockchain technology can contribute to smart cities," *Financial Innovation*, vol.2, pp.26, 2016.

[33] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk

Control," *J. Med. Syst.*, vol.40, pp.218, 2016.
[34] Q. Xia, E. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, pp. 44, 2017.
[35] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain," *IEEE Access*, vol. 5, pp. 14757-14767, 2017.
[36] V. Hassija, V. Chamola, S. Garg, D. N. G. Krishna, G. Kaddoum, and D. N. K. Jayakody, "A Blockchain-Based Framework for Lightweight Data Sharing and Energy Trading in V2G Network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5799-5812, 2020.
[37] S. Son, D. Kwon, S. Lee, Y. Jeon, A. K. Das, and Y. Park, "Design of Secure and Lightweight Authentication Scheme for UAV-Enabled Intelligent Transportation Systems Using Blockchain and PUF," *IEEE Access*, vol. 11, pp. 60240-60253, 2023.
[38] A. Vangala, A. K. Das, A. Mitra, S. K. Das, and Y. Park, "Blockchain-Enabled Authenticated Key Agreement Scheme for Mobile Vehicles-Assisted Precision Agricultural IoT Networks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 904-919, 2023.
[39] X. Zhang and X. Chen, "Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network," *IEEE Access*, vol. 7, pp. 58241-58254, 2019.
[40] J. Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Maharjan, Shengli Xie, and Yan Zhang, "Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660-4670, 2019.
[41] K. Fan, Q. Pan, K. Zhang, Y. Bai, S. Sun, and H. Li, "A Secure and Verifiable Data Sharing Scheme Based on Blockchain in Vehicular Social Networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5826-5835, 2020.
[42] J.R. Clavin, Y. Huang, X. Wang, P.M. Prakash, Sisi Duan, Jianwu Wang, S. Peisert, "A Framework for Evaluating BFT," *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 193-200, 2021, doi: 10.1109/ICPADS53394.2021.00030.
[43] T. Jiang, H. Fang, and H. Wang, "Blockchain-based internet of vehicles: Distributed network architecture and performance analysis," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4640–4649, 2019.

Mingming Cui received a B.S. degree in Computer Science and Technology from the Anhui University of Finance and Economics, China. She is currently pursuing a Ph.D. degree from Shanghai Maritime University, China, and a Visiting Ph.D. student at the Nanyang Technological University, Singapore. Her research interests include cryptology, blockchain, data privacy protection, network security, VANETS security, and the Internet of things.

Dezhi Han (Senior Member, IEEE) received a B.S. degree in applied physics from the Hefei University of Technology, Hefei, China, in 1990 and the M.S. and Ph.D. degrees in computing science from the Huazhong University of Science and Technology, Wuhan, China, in 2001 and 2005, respectively. He is currently a Professor at the Department of Computer, Shanghai Maritime University, Pudong, China, 2010. His current research interests include cloud and outsourcing security, wireless communication security, and network and information security.

Han Liu received the M.S. and Ph.D. degrees from Shanghai Maritime University, where he is currently a postdoctoral fellow. His main research interests include blockchain, IoT, cloud security, and machine learning.

Kuan-Ching Li (Senior Member, IEEE) received a Ph.D. and M.S. in electrical engineering and Licenciatura in mathematics from the University of São Paulo, Brazil, in 2001, 1996, and 1994, respectively. He is currently a Distinguished Professor at the Department of Computer Science and Information Engineering, Providence University. He received distinguished and chair professorships from universities in several countries and was the recipient of awards and funding support from several agencies and high-tech companies. He has been involved actively in many major conferences and workshops as a program/general/steering conference chairman and has organized numerous conferences/workshops. Besides publishing numerous research papers and articles, he is co-author/co-editor of several technical professional books published by CRC Press/Taylor & Francis, Springer, McGraw-Hill, and IGI Global. His research interests include parallel and distributed processing, Big Data, Blockchain, and emerging technologies. He is a member of the AAAS, a Life Member of the TACC, and a Fellow of the IET.

Mingdong Tang (Member, IEEE) received a B.S. in electrical engineering from Tianjin University, China, in 2000, a M.S. in control engineering from Shanghai University in 2003, and a Ph.D. from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2010. He is currently a professor at the School of Information Science and Technology at Guangdong University of Foreign Studies, China. His research interests include service-oriented computing, software engineering, and data mining.

Chin-Chen Chang (Fellow, IEEE) received a Ph.D. degree in computer engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1982, and the B.E. and M.E. degrees in applied mathematics, computer and decision sciences from National Tsinghua University, Hsinchu, Taiwan, in 1977 and 1979, respectively. He was with National Chung Cheng University, Taiwan, from 1989 to 2005. He has been a Chair Professor at Feng Chia University, Taiwan, since 2005. He has been invited to serve as a Visiting Professor, a Chair Professor, an Honorary Professor, the Honorary Director, the Honorary Chairman, a Distinguished Alumnus, a Distinguished Researcher, and a Research Fellow by universities and research institutes. His research interests include database design, computer cryptography, image compression, and data structures. Prof. Chang was a recipient of many research awards and honorary positions by and in prestigious organizations both nationally and internationally, such as the Outstanding Talent in Information Sciences of China, He is a Fellow of the IET and AAIA.

Ferheen Ayaz (Graduate Student Member, IEEE) completed her PhD in blockchain-based security of vehicular networks with the University of Sussex, Brighton, UK, in 2022. She received her B.E. and M.E. degrees from the NED University of Engineering and Technology, Karachi, Pakistan, in 2010 and 2014, respectively. She is currently a Research Fellow at the University of Sussex, working on 5G-enabled communications of electric vehicles to achieve a net-zero target. Previously, she was a Research Associate at University of Glasgow, Glasgow, UK, working on the security and optimization of machine learning. Her research interests include vehicular networks, blockchain applications, machine learning, and security. She was a recipient of the N2Women Fellowship 2020, IEEE Industrial Placement 2021, and runner up of the Equity, Diversity and Inclusion Rising Star Award 2022. She actively works as an organizing or technical committee member for renowned international conferences and volunteers for N2Women, IEEE UK and Ireland Women in Engineering and IEEE UK and Ireland Cybersecurity Group.

Zhengguo Sheng (Senior Member, IEEE) received the B.Sc. degree from the University of Electronic Science and Technology of China, China, in 2006, and the M.S. and Ph.D. degrees from Imperial College London, London, U.K., in 2007 and 2011, respectively. He is currently a Reader with the University of Sussex, U.K. Previously, he was with UBC, Vancouver, BC, Canada, as a Research Associate and with Orange Labs as a Senior Researcher. He has more than 130 publications. His research interests cover IoT, vehicular communications, and cloud/edge computing. He is also the receipt of Royal Society Kan Tong Po International Fellowship 2020, Emerging research award 2017 from University of Sussex. Senior Member of IEEE, IET, Fellow of The Higher Education Academy (HEA).

Yong Liang Guan (Senior Member, IEEE) obtained his PhD degree from the Imperial College London, UK, and Bachelor of Engineering degree with first class honours from the National University of Singapore. He is now a Professor of Communication Engineering at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore, where he leads the Continental-NTU Corporate Research Lab and led the successful deployment of the campus-wide NTU-NXP V2X Test Bed. His research interests broadly include coding and signal processing for communication systems and data storage systems. He has published an invited monograph, 2 books and more than 450 journal and conference papers. He has secured over S$70 million of external research funding. He has 15 filed patents and 3 granted patents, one of which was licensed to NXP Semiconductors. He is an Editor for the IEEE Transactions on Vehicular Technology. Currently he is also an Associate Vice President of NTU and a Distinguished Lecturer of the IEEE Vehicular Technology Society.