Blockchain-Based Lightweight Certificate Authority for Efficient Privacy-Preserving Location-Based Service in Vehicular Social Networks

Huajie Shen, Jun Zhou, Zhenfu Cao, *Senior Member, IEEE*, Xiaolei Dong,
and Kim-Kwang Raymond Choo, *Senior Member, IEEE*

***Abstract*—Blockchain can be utilized to enhance both security and efficiency for location-based service (LBS) in vehicular social networks (VSNs), due to its inherent decentralization, anonymity, and trust properties. Unfortunately, the existing approaches either lack effective authentication, which is vulnerable to the man-in-the-middle attack, or require an online certificate authority (CA) where frequent interactions with resource-constrained vehicles are required. To address these challenging issues, in this article, a lightweight threshold CA for consortium blockchain along with a privacy-preserving LBS protocol in blockchain enforced VSNs is proposed. First, a lightweight threshold CA framework LTCA is proposed by devising a threshold proxy signature, where the proxy signing key is issued by a coalition of threshold number of CAs playing the roles of authorized nodes in the consortium blockchain. Without the intervene of an online CA, each vehicle in the online phase can authenticate its identity by itself each time its blockchain address (i.e., account address) is updated. Then, based on the proposed LTCA, an efficient privacy-preserving LBS protocol PPVC is contrived to protect each vehicle's conditional identity privacy with a moderate cost. Finally, both security analysis and performance evaluation demonstrate the effectiveness and efficiency of our proposed LTCA and PPVC in blockchain enforced VSNs.**

***Index Terms*—Consortium blockchain, lightweight certificate authority (CA), location-based service (LBS), privacy preserving, vehicular social networks (VSNs).**

I. Introduction

With rapid increase of vehicles, a significant amount of data associated with types of vehicular services have been unprecedentedly generated, and how to intelligently gather and process those valuable data becomes a challenging issue. As an integration of the social network and Internet of Vehicles (IoV), vehicular social networks (VSNs) aim at supporting location-based services (LBSs), by reducing the increasing number of road accidents, traffic congestion, and providing valuable life information in the neighborhood, through intelligently gathering and processing kinds of vehicular communication data with mutual support [1]. Unfortunately, owing to the large scale of transmitted data, the frequent mobility and the intermittent routes among resource-constrained vehicles, both the delivery ratio and average delay for LBS information would be considerably affected and it becomes difficult to achieve realtime control and response for VSNs. Therefore, a set of appropriately assigned roadside units (RSUs) serving as fog nodes, would facilitate authorized data access and immediate vehicular communications [2].

On the other hand, VSNs are vulnerable to types of security threats where both resource-constrained vehicles and RSUs tend to be compromised by adversaries [3]. A set of corrupted vehicles aims to extract the true identity and even the driving routes of honest vehicular users. A malicious RSU would hamper vehicular communication for in VSNs, by ruining LBS data authorization and delivery. Chen *et al.* [12] proposed an efficient pseudonym changing scheme for location privacy protection in vehicular *ad hoc* networks (VANETs), by periodically updating each vehicle's pseudonym and its associated public-key certificates. However, the high-computational cost and communication cost for frequent pseudonym updating are intolerable by resource-constrained vehicles and the lack of an effective data synchronization mechanism disables the realtime LBS information dissemination in VSNs.

Blockchain, being viewed as an unmodifiable and distributed database where all blockchain nodes follow a predefined consensus to achieve the coherence and synchronization for decentralized data storage, provides an efficient solution to the immediate vehicular communication for LBS in a distributed and large-scale VSNs. Sharma *et al.* [13] presented a distributed blockchain-based vehicular network architecture in smart city, to enhance the delivery ratio in vehicular communications. Although the performance is improved compared to [12], it can still not achieve a satisfactory efficiency, since a vehicular user is adopted as a miner in the blockchain where its constrained computing power becomes a heel dragging factor. On the contrary, the technique of consortium blockchain would be favorable in many business applications. Unlike a conventional blockchain, its consensus process is completed more efficiently by a preselected set

Manuscript received October 20, 2019; revised January 13, 2020; accepted February 11, 2020. Date of publication February 19, 2020; date of current version July 10, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61602180, Grant 61702187, Grant 61632012, and Grant 61672239; in part by the Shanghai Sailing Program under Grant 17YF1404400; and in part by the China Postdoctoral Science Foundation under Grant 2017M611502. *(Corresponding author: Jun Zhou.)*

Huajie Shen, Jun Zhou, Zhenfu Cao, and Xiaolei Dong are with the Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China (e-mail: adijeshen@gmail.com; jzhou@sei.ecnu.edu.cn; zfcao@sei.ecnu.edu.cn; dongxiaolei@sei.ecnu.edu).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Digital Object Identifier 10.1109/JIOT.2020.2974874

of authorized nodes. These authorized nodes also have the obligation to register new legitimate users and to audit their misbehavior for accountability when a dispute takes place. For vehicular communications, RSU is a convincing candidate for the miner nodes in blockchain enforced VSNs to provide LBSs. Kang *et al.* [14] devised a blockchain for secure and efficient data sharing in vehicular edge computing and networks by utilizing RSUs as miners in blockchains, and hence a higher efficiency is obtained than the scheme [13] where vehicles serve as miners. Unfortunately, to achieve identity privacy for vehicular users, it is required to frequently interact with an online certificate authority (CA) to update the public-key certificates for periodically updated pseudonyms. Therefore, the heavy computational cost and communication bandwidth requirement would not only damage efficient LBS content delivery but also deviate from the lightweight features for resource-constrained vehicles.

More significantly, all the existing works [12]–[14] are vulnerable to types of security and privacy threats. Recent work [4], [15], [24]–[26] has shown a background analysis attack where the authentic IP address of a blockchain user would be successfully inferred with high probability, by analyzing a series of publicly accessible historical transaction data associated to a specific blockchain address. Furthermore, owing to the fact that a pseudorandom string is adopted as the blockchain user's address lacking the linkability to its true identity, the accountability cannot be achieved by effectively tracing misbehavior without anyone knowing their true identities. In VSNs, conditional identity privacy is required that a users' identity privacy should be well protected unless misbehavior or dispute is detected. More critically, without the linkability between the user's true identity and its account address as pseudorandom strings, an adversary can replace an honest blockchain user's address as his own forged one to launch the man-in-the-middle attack.

To address these challenging issues, in this article, a lightweight threshold CA LTCA for consortium blockchain and an efficient privacy-preserving LBS protocol PPVC in blockchain enforced VSNs have been proposed. The main contributions are presented as follows.

1) A threshold proxy signature scheme is designed, based on which a lightweight threshold CA LTCA for consortium blockchain is proposed to resist man-in-the-middle attack. Specifically, a coalition of the threshold number of CAs is required to generate a proxy signing key only once for the legitimate user in the offline phase, which can be utilized to authenticate the vehicle's identity by itself every time an updated account address is generated in the online phase without incurring frequent interactions with the intervention of an online CA.
2) Based on our proposed LTCA, an efficient privacy-preserving LBS protocol PPVC is presented. It effectively resists the background analysis attack by frequently updating the vehicle's blockchain address and achieves the accountability by conditionally linking the blockchain address to the user's true identity.
3) Security analysis and extensive performance evaluations demonstrate the effectiveness and the efficiency of our proposed schemes, especially the conditional identity privacy is achieved by implementing an offline CA with $O(1)$ complexity, independent of the number of account addresses, and high efficiency is achieved by constructing a consensus protocol based on the proof-of-stake protocol.

The remainder of this article is organized as follows. In Section III, we describe the network architecture and the consensus process. In Section IV, the protocol definitions and security requirements are demonstrated. In Section V, a lightweight threshold CA framework LTCA is proposed, followed by an efficient privacy-preserving LBS protocol PPVC is given in Section VI. In Section VII, both security analysis and performance evaluations are presented. Finally, we conclude this article in Section VIII.

II. Related Work

Due to the current interest in blockchain, there have been efforts to leverage the decentralized, anonymous, and trustable features of conventional blockchains to address security and privacy issues in distributed networks [5]–[8], [13], [14], [21]. Rowan *et al.* [5] devised secure vehicle-to-vehicle communication using blockchain through visible light and acoustic side-channels. Sharma *et al.* [13] presented a distributed blockchain-based vehicular network architecture in a smart city. Aitzhan and Svetinovic [10] achieved secure energy trading in a peer-to-peer (P2P) manner by exploiting the techniques of multisignatures, blockchain, and anonymous messaging streams. Unfortunately, owing to the restriction of conventional blockchain techniques, the state of the art mentioned above are vulnerable to the man-in-the-middle attack. Specifically, since most blockchain addresses are pseudorandom strings generated by Base58 encoding, which are hard for human beings to remember, blockchain-based vehicular users would always copy-and-paste the receivers' addresses for sending messages. However, receiver's blockchain addresses are not effectively bound to their true identities and an adversary can trivially replace them with its forged one without being detected. For example, CryptoCurrency Clipboard Hijacker [22] has been reported to monitor 2.3 million cryptocurrency addresses and successfully launch man-in-the-middle attacks resulting in the loss of millions of dollars. If it were applied to LBS in VSNs, a large quantity of unauthorized information leakage would take place.

To address this issue, most of the existing approaches [14], [16], [17], [19] resist man-in-the-middle attacks and achieve accountability, by introducing a centralized CA (e.g., a government department), which takes charge of assigning public-key certificates for registered vehicles. The need of a centralized CA, however, contradicts the distributed architecture of blockchain enforced VSNs design and tends to suffer from single node failure. Additionally, to achieve conditional identity privacy in blockchain [11], [23] and resist background analysis attacks [4], [5], [24]–[26], a straightforward approach is to assign a set of account addresses (i.e., pseudonyms) to each user for periodical

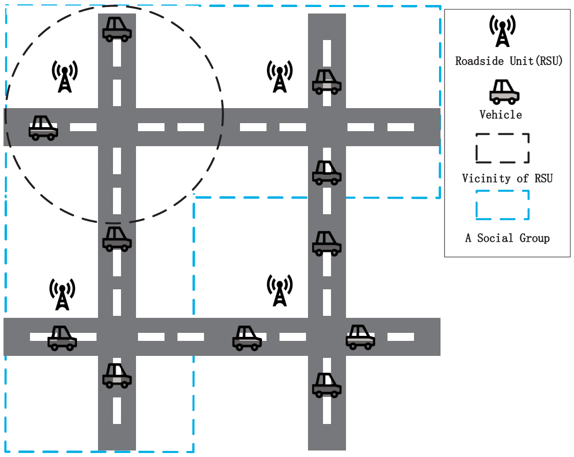


Fig. 1. Network architecture of blockchain enforced VSNs.

updating and an online CA is needed to update the corresponding certificate every time a new blockchain address is adopted. Unfortunately, the frequent interactions between CA and resource-constrained vehicle-carried computer would incur considerably heavy complexities. Recently, Sun *et al.*'s RingCT 2.0 protocol [18] used a linkable ring signature scheme to protect the user's identity privacy. However, the computational cost significantly increases as the number of groups of input accounts increases. Chen *et al.* proposed CertChain [27], where a blockchain-based decentralized CA was constructed. However, in this approach, users still have to interact with an online CA each time a certificate is needed. By the way, how to store a copy of blockchain locally at the resource-constrained vehicle's end is still unsolved.

Distinguishing from the above-mentioned state of the art, in this article, a set of threshold number RSUs in blockchain enforced VSNs serving as distributed CAs can generate a proxy signing key only once in the offline phase, for the legitimate user to authenticate its identity by itself every time the account address is updated in the online phase without incurring frequent interactions with the intervention of an online CA. The linkability between the user's real identity and its pseudorandom account address, privately kept by RSUs, guarantees the accountability, the conditional identity privacy, and effectively resists man-in-the-middle attack.

III. Network Architecture and Consensus Protocol

A. Network Architecture

The integrity of blockchain and VSNs provides efficient LBSs in a large scale of vehicles in a decentralized manner. Fig. 1 shows the network structure of our proposed blockchain enforced VSNs. In blockchain enforced VSNs, the black dotted circle denotes the vicinity of each RSU. A set of RSUs that is geographically adjacent constitutes a vehicular social group, which is represented by the blue dotted square in Fig. 1. To enhance the scalability, each vehicle is connected to a nearby RSU which records and filters the LBS messages from illegal vehicles and transmits the authenticated ones to all other RSUs and vehicles located in the same vehicular social group. Compared to the architecture of blockchain-based VANET Sharma *et al.* presented in [13] regarding a single vehicle as a miner, we adopt RSUs with more computation power and storage space as the miner (i.e., playing the role of decentralized CAs as authorized nodes) in the consortium blockchain, where the computationally intensive task of transmitting flooding LBS messages and the executing consensus process would be relieved from the end of resource-constrained vehicles.

All RSUs in a vehicular social group are considered to be miners in the blockchain enforced VSNs. A PoS-like consensus protocol described in Section III-C is adopted to select the miner who would successfully generate a new block. The more active an RSU is considered, the more likely it would be selected as the leader generating a new block.

For data consistency and storage efficiency concern, each vehicular social group maintains one blockchain. The vehicular communication data are highly location sensitive and all RSUs in the same social group collect similar communication data, therefore it is reasonable that the vehicular communication data in one social group should be stored in a single chain. On the contrary, if each RSU maintains a single chain, the data consistency of one vehicular social group cannot be archived; if a single chain is maintained for all RSUs, the data redundancy would be high.

On the other hand, to prevent vehicles' identity privacy against unauthorized entities, vehicles' short-term public key (blockchain address) and the corresponding public-key certificates are required to be updated periodically or every time they enter a new VSN, to resist both the background analysis attack and the man-in-the-middle attack.

Last but not least, the RSUs in VSNs serving as miners and decentralized CAs for consortium blockchain would guarantee the LBS messages recorded synchronously and disseminated correctly among all regions different RSUs take responsibility of within the same vehicular social group. Compared to [12] and [13] where resource-constrained vehicles take charge of LBS message delivery in a VSNs of high mobility and intermittent routing, both the message delivery ratio and the average delay would be optimized.

B. System Model

Blockchain enforced VSNs provide an effective and secure way of location-based vehicular communication. Fig. 2 demonstrates the system model of blockchain enforced VSNs composed of vehicles and RSUs. Vehicles are common users and each of them possesses a pseudonym, a corresponding long-term public-key/secret-key pair and a short-term public key (blockchain address) that would be updated afterward. RSUs are authorized nodes selected as trustworthy members in the initialization of blockchain enforced VSNs, in each of which, a transaction server is utilized to provide LBS communication services and records all LBS messages into a memory pool (blockchain). Besides, an account pool is adopted to store vehicular users' true identities, their associated long-term public keys and the corresponding limits of authorization (including the pseudonym). Fig. 2 also presents

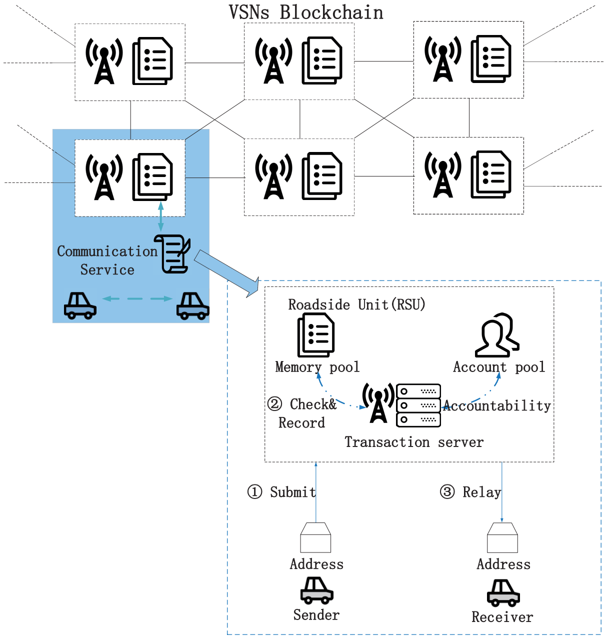


Fig. 2. System model of blockchain enforced VSNs.

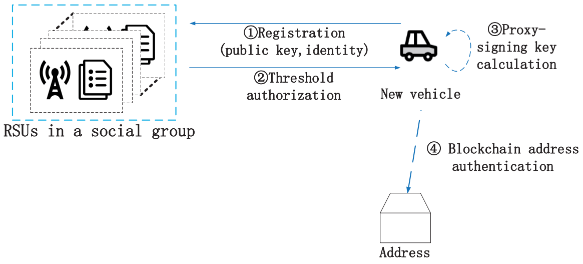

①Registration (public key, identity)
②Threshold authorization
③Proxy-signing key calculation
New vehicle
RSUs in a social group
④ Blockchain address authentication
Address

Fig. 3. Vehicle authorization and authentication.

the process of LBS message delivery between vehicles in blockchain enforced VSNs as follows.

1) When a vehicle sender enters a new social group, it updates its short-term public key and generates a certificate on the public key, then it publicizes a tuple of the pseudonym, the short-term public key and the corresponding certificate on the public bulletin.
2) The vehicle sender decides the receiver according to its pseudonym, verifies the certificate and generates and submits an encrypted LBS message under the receiver's short-term public key to its nearest RSU if the certificate is valid.
3) The RSU checks the validity of the encrypted LBS message and records it into the memory pool if it is successfully verified.
4) The RSU relays the encrypted LBS message to the receiver who can successfully perform decryption using its secret key. Note that in our system model, most calculation and storage burden to run the consensus protocol are loaded onto powerful RSUs rather than resource-constrained vehicles, which enables to improve the scalability of the blockchain enforced VSNs.

A set of n selected RSUs also serves the functionality of decentralized CAs. Fig. 3 shows the authorization and authentication process of vehicles as follows: 1) when a new vehicular user registers its public-key information to blockchain enforced VSNs, she/he would be authenticated in a trusted manner (i.e., it is required to upload the driving license and ID card to the RSUs); 2) each RSU validates the user's information and issues a partial proxy signing key to it; 3) holding no less than the threshold number of partial proxy signing keys, the vehicular user can retrieve a complete proxy signing key; and 4) with the proxy signing key, each vehicular user can generate the certificates of her/his temporary public keys associated to the updated blockchain addresses by her/himself without the intervene of RSUs serving as decentralized CAs.

TABLE I
LBS COMMUNICATION DATA STRUCTURE

Hashcode:$H(C)$	
Sender's address: Adr_a	Receiver's address:Adr_b
Encrypted Message: c_m	
Script: $\sigma_{c_m}, (pk_a, Cert_a), (pk_b, Cert_b)$	
Other Data: Previous Hash, Time Stamp, Merkle Root	

The LBS communication data structure of blockchain enforced VSNs in our design is demonstrated in Table I. It shows an example of an LBS delivery between sender a and receiver b. The data structure contains $H(C)$, a hashcode of the underlying LBS communication C, two vehicles' blockchain addresses Adr_a, Adr_b, and a ciphertext c_m of message m. The transaction script contains a signature σ_{c_m} of the encrypted message c_m, sender a and receiver b's public keys and their associated certificates $(pk_a, Cert_a), (pk_b, Cert_b)$. It also includes other data, such as the hashcode of the previous LBS communication, the timestamp, and the Merkle root.

C. Consensus Protocol

In our blockchain enforced VSNs, owing to the resource constraint of vehicles, a consortium blockchain is exploited, where RSUs run the consensus protocol and store a copy of the complete blockchain containing all LBS communication messages in each of their memory pools, to which the authorized vehicles can successfully access. Our consensus protocol is constructed, by modifying Ouroboros [**?**] with the exception that the stake is replaced by a rank of activity that denotes RSU's LBS message relay participation rate in the *Leader Slection* process.

Leader Selection: RSU$_i$ is selected to be the leader among n RSUs with probability p_i of its rank of activity, where $p_i = (a_i/[\sum_{j=1}^{n} a_j])$, and $a_j (j = 1, 2, \ldots, n)$ denotes the times of RSU$_j$ successfully verifies vehicular LBS messages and relays the messages to the receiver.

Incentive Mechanism: Each time an RSU relays an LBS message, its rank of activity increases by one. It is believed a higher rank of activity usually means a higher availability and less effort to receive/transmit LBS messages from/to other RSUs.

Chain Extension: For each epoch in the life cycle of blockchain enforced VSNs, each RSU performs the following steps.

1) Each RSU collects all valid blockchains and verifies all blocks in the longest valid blockchain. If all verification results are true, it adopts the longest valid blockchain as the main blockchain.
2) If an RSU is selected as the leader in the current epoch, it generates a new block after checking all vehicular LBS communication data in this epoch. It adds the new block into the current blockchain and broadcasts it to all other RSUs. Then, its rank of activity is reset to zero to achieve the fairness in leader selection.

IV. Protocol Definition and Security Requirement

A. Protocol Definitions

Definition 1: The proposed lightweight threshold CA framework for consortium blockchain consists of the following five algorithms.

1) $CAK-GEN(1^{\lambda}) \rightarrow (pk_{CA}, sk_{CA})$*:* CA's key generation, a probabilistic polynomial time (PPT.) algorithm run by system that takes the security parameter 1^{λ} as input and outputs CA's public/secret-key pair (pk_{CA}, sk_{CA}).
2) $CAK-DST(1^{\lambda}, sk_{CA}) \rightarrow (d_1, d_2, \ldots, d_n)$*:* CA's key distribution, a deterministic algorithm run by the system that takes the input 1^{λ} and sk_{CA}, and outputs n partial secret key d_is, each of which is sent to RSU$_i$, respectively, (noted that RSU$_i$ is one of n static RSUs of blockchain enforced VSNs).
3) $ULK-GEN(1^{\lambda}) \rightarrow (pk_u, sk_u)$*:* User's long-term key generation, a PPT. algorithm run by vehicle u that takes the input 1^{λ} and outputs a long-term public/secret-key pair (pk_u, sk_u).
4) $PSK-GEN(1^{\lambda}, d_i, pk_u, m_{\omega u}) \rightarrow (v_{u,i})$*:* Partial proxy-signing key generation, a PPT. algorithm run by RSU$_i$ that takes the inputs 1^{λ}, d_i and pk_u and the limits of authorization (including the pseudonym) $m_{\omega u}$, and outputs the partial proxy-signing-key $v_{u,i}$ and sends it to vehicle u.
5) $PSK-REC(1^{\lambda}, pk_{CA}, v_{u,i_1}, \ldots, v_{u,i_k}) \rightarrow (v_u)$*:* Proxy-signing key recovery, a deterministic algorithm run by vehicle u that takes the inputs 1^{λ}, pk_{CA} and $v_{u,i_j}(i_j \in i_1, i_2, \ldots, i_k)$ (k is the threshold number of key recovery), and outputs a proxy-signing key v_u.

Definition 2: The proposed privacy-preserving LBS protocol in blockchain enforced VSNs consist of the following seven algorithms.

1) $BCA-INI(1^{\lambda}) \rightarrow (pk_{b,j}, sk_{b,j})$*:* Blockchain address's initialization, a PPT. algorithm run by receiver b (sender a) [same as u, writing as b (a) in Definition 2 to better understand the relation between vehicular user a and b, assuming that a wants to send some message to b] that takes the input 1^{λ} and outputs a public/secret-key pair $(pk_{b,j}, sk_{b,j})(j = 1, \ldots, l)$ (j denotes times b updates his address; note that only one key pair in one call to $BCA-INI$) and an address of wallet $Adr_{b,j}$.
2) $CRT-GEN(1^{\lambda}, m_{\omega b}, v_b, pk_{b,j}, pk_{CA}, sk_b, pk_b) \rightarrow (Cert_{b,j})$*:* Certificate generation, a PPT. algorithm run by receiver b (sender a) that takes the inputs $1^{\lambda}, m_{\omega b}, v_b, pk_{b,j}, pk_{CA}, sk_b, pk_b$, and outputs the public certificate $Cert_{b,j}$ on $pk_{b,j}$.
3) $BCA-VER(1^{\lambda}, pk_{b,j}, Cert_{b,j}, pk_{CA}, pk_b) \rightarrow (\{1, \perp\})$*:* Blockchain address's verification, a deterministic algorithm run by sender a that takes the inputs 1^{λ}, $(pk_{b,j}, Cert_{b,j})$, pk_{CA} and pk_b, and outputs 1 if $Cert_{b,j}$ is valid (i.e., the address receiver b is not altered by some malware), otherwise, it outputs $\perp$.
4) $MSG-ENC(1^{\lambda}, m, pk_{b,j}) \rightarrow (c_m)$*:* Message's encryption, a PPT. algorithm run by sender a that takes the inputs 1^{λ}, the message m and $pk_{b,j}$, and outputs the ciphertext c_m of m.
5) $CPT-DEC(1^{\lambda}, c_m, sk_{b,j}) \rightarrow (m)$*:* Ciphertext's decryption, a deterministic algorithm run by receiver b that takes the inputs 1^{λ}, the ciphertext c_m and $sk_{b,j}$, and outputs the message m.
6) $MSG-SIG(1^{\lambda}, c_m, pk_{a,j'}, sk_{a,j'}) \rightarrow (\sigma_{c_m})$*:* Message's signature, a PPT. algorithm run by sender a that takes the inputs 1^{λ}, the ciphertext c_m and $sk_{a,j'}$, and outputs the signature σ_{c_m} of ciphertext c_m.
7) $SIG-VER(1^{\lambda}, \sigma c_m, c_m, pk_{a,j'}, Cert_{a,j'}, pk_{CA}) \rightarrow (\{1, \perp\})$*:* Signature's verification, a deterministic algorithm run by RSU that takes the inputs 1^{λ} and σc_m, c_m, $pk_{a,j'}$ and pk_{CA}, and outputs 1 if σ_{c_m} is valid, otherwise, outputs $\perp$.

B. Security Requirements

The proposed lightweight threshold CA for consortium blockchain and the privacy-preserving LBS protocol in blockchain enforced VSNs aims to achieve the following security goals.

1) *Conditional Identity Privacy Preserving:* Each vehicular user's identity is required to be protected against others. On the other hand, to achieve accountability, RSU should be able to trace the vehicular user's identity once some abnormal behavior or vehicle's misbehavior are detected.
2) *Man-in-the-Middle-Attack Resistance:* Man-in-the-middle-attack refers to a dishonest vehicular user (adversary) who can deceive the other vehicle to send message to her/his address by replacing the former receiver's address by her/his forged one, since the binding relationship between vehicular user's true identity and blockchain address cannot be guaranteed. To resist this kind of attack, our scheme aims to achieve certificate's existential unforgeability under an adaptively chosen message attack (EUF-CMA). The security model is formally defined as follows.

Definition 3: Let $\Pi_{Cert} = (CAK-GEN, CAK-DST, ULK-GEN, PSK-GEN, PSK-REC, BCA-INI, CRT-GEN, BCA-VER)$ be a lightweight threshold certificate generation and verification scheme (i.e., refer to Sections V and VI for details), and let $\mathcal{A}$ be the collusion between a PPT forger and a coalition of corrupted RSUs S^{Cpt} of size $|S^{Cpt}| < k$. For $1^{\lambda} \in \mathbb{N}, i \in \mathbb{K}(|\mathbb{K}| = n \geq k)$ where $\mathbb{K}$ and k, respectively, denote the set of all RSUs and the threshold for RSU corruption, let

$$Adv^{EUF-CMA}_{\mathcal{A}_1,\Pi_{Cert}}(1^\lambda, t, q_c) = \Pr\left[\begin{array}{l} (sk_{CA}, pk_{CA}) \leftarrow CAK-GEN(1^\lambda) \\ (d_i) \leftarrow CAK-DST(1^\lambda, sk_{CA}) \\ (pk_u, sk_u) \leftarrow ULK-GEN(1^\lambda) \\ v_{u,i} \leftarrow PSK-GEN(1^\lambda, d_i, pk_u, m_{\omega u}) \\ v_u \leftarrow PSK-REC(1^\lambda, v_{u,i}, pk_{CA}) \\ (pk_{u,j}, sk_{u,j}) \leftarrow WAL-INI(1^\lambda) \\ (pk_{\mathcal{A}_1,j}, Cert_{\mathcal{A}_1,j}) \leftarrow \mathcal{A}_1^{\mathcal{O}_1} \\ \quad \times (pk_{u,j}, pk_{CA}, pk_u, v_u, m_{\omega u}) \\ BCA-VER(pk_{\mathcal{A}_1,j}, Cert_{\mathcal{A}_1,j}, pk_u, pk_{CA}) = 1 \\ \wedge pk_{\mathcal{A}_1,j} \notin \mathcal{M}_{\mathcal{O}_1} \end{array}\right]$$

where $\mathcal{O}_1$ denotes the certificate generation oracle and hash oracles, $\Pr[\cdot]$ denotes the probability, t denotes the running time, and q_c denotes the most query times performed by the adversary $\mathcal{A}_1$ to the certificate generation oracle and hash oracles $\mathcal{O}_1$; while $\mathcal{M}_{\mathcal{O}_1}$ is the set of all public keys that have been submitted to the certificate generation oracle ($CRT-GEN$) during $\mathcal{A}_1$'s whole querying process.

The proposed Π_{Cert} is EUF-CMA secure if

$$Adv^{EUF-CMA}_{\mathcal{A}_1,\Pi_{Cert}}(1^\lambda, t, q_c) < \epsilon(1^\lambda).$$

3) *LBS Message's Signature Unforgeability:* An adversary cannot forge a signature of an LBS message on behalf of an authentic message sender. Otherwise, anyone can fabricate an LBS message and lead to a mess.

Definition 4: Let $\Pi_{Sig} = (CAK-GEN, CAK-DST, ULK-GEN, PSK-GEN, PSK-REC, BCA-INI, MSG-SIG, SIG-VER)$ be a location-based vehicular communication's signature protocol (i.e., refer to Sections V and VI for details), and let $\mathcal{A}$ be a PPT forger. For $1^\lambda \in \mathbb{N}$, let

$$Adv^{EUF-CMA}_{\mathcal{A}_2,\Pi_{Sig}}(1^\lambda, t, q_c) = \Pr\left[\begin{array}{l} (sk_{CA}, pk_{CA}) \leftarrow CAK-GEN(1^\lambda) \\ (d_i) \leftarrow CAK-DST(1^\lambda, sk_{CA}) \\ (pk_u, sk_u) \leftarrow ULK-GEN(1^\lambda) \\ v_{u,i} \leftarrow PSK-GEN(1^\lambda, d_i, pk_u, m_{\omega u}) \\ v_u \leftarrow PSK-REC(1^\lambda, v_{u,i}, pk_{CA}) \\ (pk_{u,j}, sk_{u,j}) \leftarrow WAL-INI(1^\lambda) \\ (\sigma_{c_m}) \leftarrow \mathcal{A}_2^{\mathcal{O}_2}(pk_{u,j}, pk_{CA}, pk_u, v_u, c_m) \\ SIG-VER(\sigma_{c_m}, pk_u, pk_{CA}, c_m) = 1 \\ \wedge c_m \notin \mathcal{M}_{\mathcal{O}_2} \end{array}\right]$$

where $\mathcal{O}_2$ denotes the signature generation oracle and hash oracles, $\Pr[\cdot]$ denotes the probability, t denotes the running time, and q_c denotes the most query times performed by the adversary $\mathcal{A}_2$ to the signature generation oracle and hash oracles $\mathcal{O}_2$; while $\mathcal{M}_{\mathcal{O}_2}$ is the set of all public keys that have been submitted to the signature generation oracle ($MSG-SIG$) during $\mathcal{A}_2$'s whole querying process.

The proposed Π_{Sig} is EUF-CMA secure if

$$Adv^{EUF-CMA}_{\mathcal{A},\Pi_{Sig}}(1^\lambda, t, q_c) < \epsilon(1^\lambda).$$

4) *LBS Message Confidentiality:* Only the authorized receivers can successfully decrypt LBS messages.

Definition 5: Let $\Pi_{Enc} = (CAK-GEN, CAK-DST, ULK-GEN, PSK-GEN, PSK-REC, BCA-INI, MSG-ENC, CPT-DEC)$ be a location-based vehicular communication's message encryption protocol (i.e., refer to Sections V and VI for details), and let $\mathcal{A}_3$ be a PPT adversary. For $1^\lambda \in \mathbb{N}$, let

$$Adv^{ind-cca-atk}_{\mathcal{A}_3,\Pi_{Enc}}(1^\lambda, t, q_c) = \left|\Pr\left[\begin{array}{l} (sk_{CA}, pk_{CA}) \leftarrow CAK-GEN(1^\lambda) \\ (d_i) \leftarrow CAK-DST(1^\lambda, sk_{CA}) \\ (pk_u, sk_u) \leftarrow ULK-GEN(1^\lambda) \\ v_{u,i} \leftarrow PSK-GEN(1^\lambda, d_i, pk_u, m_{\omega u}) \\ v_u \leftarrow PSK-REC(1^\lambda, v_{u,i}, pk_{CA}) \\ (pk_{u,j}, sk_{u,j}) \leftarrow WAL-INI(1^\lambda) \\ Cert_{u,j} \leftarrow CRT \\ \quad -GEN(1^\lambda, pk_{u,j}, v_u, pk_{CA}, m_{\omega u}) \\ (m_0, m_1) \leftarrow \mathcal{A}_3^{\mathcal{O}_3}(pk_{u,j}, Cert_{u,j}, pk_{CA}, v_u); \\ b \leftarrow \{0, 1\} \\ c^* \leftarrow MSG-ENC(m_b, pk_{u,j}, pk_{CA}) \\ b \leftarrow \mathcal{A}_3^{\mathcal{O}_3}(m_0, m_1, c^*) \wedge (c^*) \notin \mathcal{M}_{\mathcal{O}_3} \end{array}\right] - \frac{1}{2}\right|$$

where $\mathcal{O}_3$ denotes decryption oracle and hash oracles, $\Pr[\cdot]$ denotes the probability, t denotes the running time, and q_c denotes the most query times performed by the adversary $\mathcal{A}_3$ to the decryption oracle and hash oracles; while $\mathcal{M}_{\mathcal{O}_3}$ is the set of messages that have been submitted to the decryption oracle during $\mathcal{A}_3$'s whole querying process.

The proposed Π_{Enc} is ind-cca-atk secure if

$$Adv^{ind-cca-atk}_{\mathcal{A}_3,\Pi_{Enc}}(1^\lambda, t, q_c) < \epsilon(1^\lambda).$$

5) *CA's Misbehavior Detectability:* A malicious CA would disable a legitimate vehicle to authenticate himself to others by issuing an incorrect proxy signing key. This misbehavior should be detectable in our proposed scheme.

6) *Single-Point Failure Resistance:* A single CA is not only inappropriate for the decentralized architecture of the blockchain but also tends to become the attacking target of the adversary. It is required to be addressed by devising threshold CA in our scheme.

V. Lightweight Threshold CA Framework LTCA for Consortium Blockchain

In this section, a lightweight threshold CA framework LTCA for consortium blockchain is proposed by designing a threshold proxy signature scheme. The distributed CA is well fit for the consortium blockchain architecture and is secure against a malicious CA.

A notation list is presented in Table II to better understand the proposal in Sections V and VI.

A. System Initialization

CAK-GEN$(1^\lambda) \rightarrow (pk_{CA}, sk_{CA})$*:* First, the system takes a security parameter 1^λ as input and selects two large secure primes p_0 and q_0 of size λ, where $p_0 = 2p_0' + 1$, $q_0 = 2q_0' + 1$, with p_0', q_0' themselves prime. Then, it calculates $n_0 = p_0 \cdot q_0$

TABLE II
NOTATIONS

Notation	Description
$pk_{CA} = (n_0, e_0)$	The public key of CA
$sk_{CA} = d_0$	The secret key of CA
RSU_i	The i-th RSU in a VSN
n	Total number of RSUs in a VSN
k	Threshold number of RSUs for certificate
d_i	Partial secret key of d_0 held by RSU_i
$pk_u = (n_u, e_u)$	The vehicle u's long term public key, used to generate a certificate on short term public key
$sk_u = d_u$	The vehicle's long term secret key
$v_{u,i}$	Partial proxy signing key for vehicle u generated by RSU_i
v_u	Complete proxy signing key for vehicle u
$m_{\omega u}$	Some limits of authorization including the pseudonym of vehicle u
$j (j = 1, 2, \cdots, l)$	The update times of short term key pairs of vehicle b
$j' (j' = 1, 2, \cdots, l)$	The update times of short term key pairs of vehicle a
l	The total times a vehicle will update its short term key pairs
$pk_{b,j} = (n_{b,j}, e_{b,j})$	The short term public key for vehicle b, used for communication between vehicles
$sk_{b,j} = (d_{b,j})$	The short term secret key for vehicle b
$Cert_{b,j}$	Certificate of $pk_{b,j}$

as the RSA-modulus, $m_0 = p'_0 \cdot q'_0$ and d_0, e_0 as the corresponding decryption exponent and encryption exponent such that $e_0 d_0 \equiv 1(\bmod\ \varphi(m_0))$ and e_0 is required to be not larger than the output of $h(\cdot)$, where $h(\cdot) : \{0, 1\}^* \rightarrow \mathbb{Z}_{n_0}$ is a cryptographic one-way hash function. Finally, it sets $pk_{CA} = (n_0, e_0)$ as public key and keeps $sk_{CA} = d_0$ secret. Then, a set of n trustworthy nodes is selected as the RSUs.

CAK-DST $(1^\lambda, sk_{CA}) \rightarrow (d_1, d_2, \ldots, d_n)$*:* Resorting to the technique of Shamir's (n, k) secret sharing, the system splits the decryption exponent $sk_{CA} = d_0$ into n partial secret key $d_i (i = 1, 2, \ldots, n)$. The system selects a polynomial $f(x) \in \mathbb{Z}_{n_0}[x]$ of degree $k - 1$

$$f(x) = a_{k-1}x^{k-1} + \cdots + a_2 x^2 + a_1 x + d_0 (\bmod\ m_0) \tag{1}$$

where $a_i \in \mathbb{Z}_{n_0}$ $(i = 1, \ldots, k - 1)$. For each RSU RSU$_i$ $(i = 1, \ldots, n)$, the algorithm calculates $d_i = f(i)$ and sends d_i as the partial secret key to RSU$_i$. Finally, the system destroys p_0, q_0, and d_0.

Let $\Delta = n!$, for any subset S of k RSUs denoted as $i_1, \ldots, i_k$, for any $i \in \{0, \ldots, n\} \backslash S$, and $j \in S$.

Define

$$\tau^S_{i,j} = \Delta \frac{\prod_{j' \in S \backslash \{j\}} (i - j')}{\prod_{j' \in S \backslash \{j\}} (j - j')}. \tag{2}$$

From the Lagrange polynomial, it can be concluded that

$$\Delta \cdot f(i) = \sum_{j \in S} \tau^S_{i,j} \cdot f(j) \bmod m_0. \tag{3}$$

B. User's Long-Term Key Generation

ULK-GEN$(1^\lambda) \rightarrow (pk_u, sk_u)$*:* When a new vehicle u joins the blockchain enforced VSNs, the vehicular user is required to register before use. Hence, she/he will a pair of long-term key (pk_u, sk_u) for other vehicles to identify her/him. The user u takes 1^λ as input and selects two large secure prime numbers p_u and q_u of size λ. Then, it calculates $n_u = p_u \cdot q_u$ as the RSA-modulus and (d_u, e_u) as the corresponding decryption and encryption exponent such that $e_u d_u \equiv 1(\bmod\ \varphi(n_u))$ and e_u is required to be not larger than the output of $h(\cdot)$. Finally, user u sets $sk_u = (q_u, p_u, d_u)$ as the long-term secret key and $pk_u = (n_u, e_u)$ as the long-term public key.

C. Proxy Signing Key Generation and Recovery

PSK-GEN$(1^\lambda, d_i, pk_u, m_{\omega u}) \rightarrow (v_{u,i})$*:* Vehicle u calculate $x = h(m_{\omega u}, n_u, e_u)$ and sends a register message $(u, pk_u, m_{\omega u}, x)$ to all n RSUs, where u, $m_{\omega u}$, and pk_u, respectively, represents the vehicular user's true identity, the limits of authorization (including the pseudonym) and valid periods of pk_u, and the vehicle's long-term public key.

When RSU$_i$ receives the request message $(u, pk_u, m_{\omega u}, x)$, it verifies vehicular user u's information. If the vehicle and vehicular user is legitimate, it stores $(u, pk_u, m_{\omega u})$ to its account pool and calculates the partial proxy signing key $v_{u,i} = x^{2 \cdot \Delta \cdot d_i}$, and replies an acknowledgement message $v_{u,i}$ to user u.

PSK-REC $(1^\lambda,\ pk_{CA}, v_{u,i_1}, \ldots, v_{u,i_k}) \rightarrow (v_u)$*:* If a vehicle u has received no less than k acknowledgement messages [28] denoted as $\{v_{u,i_1}, v_{u,i_2}, \ldots, v_{u,i_k}\}$ from the RSU set $\{i_1, i_2, \ldots, i_k\}$, it can calculate the proxy signing key v_u as follows:

$$w = v_{u,i_1}^{2\tau^S_{0,i_1}} \cdots v_{u,i_j}^{2\tau^S_{0,i_j}} \tag{4}$$

where $v_{u,i_j}^{2\tau^S_{0,i_j}} = x^{4\Delta d_{i_j} \cdot \tau^S_{0,i_j}}$, from (3), we have $w^{e_0} = x^{e'}$, where $e' = 4\Delta^2$.

Because $(e_0, e') = 1$, we can find a pair (a, b) (where $e'a + e_0 b = -1$) and calculate $v_u = w^a x^b$.

Vehicle u can verify the correctness of v_u [whether $v_u = h(m_{\omega U}, n_U, e_U)^{-d_0}$] by checking whether $COR \equiv 1(\bmod\ m_0)$ holds, where

$$COR = v_u^{e_0} h(m_{\omega u}, n_u, e_u)(\bmod\ m_0). \tag{5}$$

If it holds, vehicle u takes v_u as the proxy signing key; otherwise, it broadcasts a complaint message to RSUs.

VI. Privacy-Preserving Location-Based Service Protocol PPVC in Blockchain Enforced VSNs

In this section, based on our LTCA, an efficient privacy-preserving LBS protocol PPVC is proposed in blockchain enforced VSNs. The advantage of the proposed scheme lies in that each vehicle can update its blockchain address and generate a certificate itself without the intervention of an online CA. Additionally, conditional identity privacy (i.e., accountability) can be well protected. For a brief explanation, we take the communication between sender a and receiver b with the detail to our protocol as follows.

A. Blockchain Address Initialization

BCA-INI $(1^\lambda) \rightarrow (pk_{b,j}, sk_{b,j})$*:* In the blockchain enforced VSNs, each vehicle holds a blockchain address to transfer and receive messages. To protect the vehicular user's identity privacy, the blockchain address is required to be periodically updated. It is assumed that an address will be periodically updated l time.

For the $j(j = 1, 2, \ldots, l)$th address creation/updating, receiver b (i.e., sender a performs the same operations) takes the security parameter 1^{λ} as input and selects two large secure prime numbers $p_{b,j}$ and $q_{b,j}$. It calculates $n_{b,j} = p_{b,j} \cdot q_{b,j}$ as the RSA-modulus and $(d_{b,j}, e_{b,j})$ as the corresponding decryption and encryption exponent such that $e_{b,j} d_{b,j} \equiv 1(\text{mod } \varphi(n_{b,j}))$ and $e_{b,j}$ should be not larger than the output of $h(\cdot)$. Then, receiver b hashes $n_{b,j}$ into an address by calculating $Adr_{b,j} = h'(n_{b,j})$ where $h'(\cdot) : \{0, 1\}^{2\lambda} \rightarrow \{0, 1\}^{\lambda}$. Finally, b keeps $sk_{b,j} = (d_{b,j})$ as the secret key, $pk_{b,j} = (n_{b,j}, e_{b,j})$ as the public key, and $(Adr_{b,j})$ as the her/his jth updated wallet's address in blockchain enforced VSNs.

B. Blockchain Address's Certificate Generation and Verification

CRT-GEN$(1^{\lambda}, m_{\omega b}, v_b, pk_{b,j}, pk_{CA}, sk_b, pk_b) \rightarrow (Cert_{b,j})$*:* When receiver b wants to receive message from sender a or other vehicles, it is required to give sender a and other vehicles its blockchain address together with its certificate. Receiver b takes the security parameter $1^{\lambda}, m_{\omega b}, v_b, pk_{b,j}, pk_{CA}, sk_b, pk_b$ as input, randomly chooses $t_j \in_R \mathbb{Z}_{n_0}$ and calculates

$$\begin{aligned} r_j &\leftarrow t_j^{e_0} (\text{mod } n_0), \quad k_j \leftarrow h\big(r_j, n_{b,j}, e_{b,j}\big) \\ g_j &\leftarrow k_j^{d_b} (\text{mod } n_b), \quad y_j \leftarrow t_j \cdot v_b^{k_j} (\text{mod } n_0) \\ Cert_{b,j} &= \big(m_{\omega b}, y_j, g_j\big). \end{aligned} \tag{6}$$

The associated certificate of public key $pk_{b,j}$ associated to the jth updated wallet address $Adr_{b,j}$ is $Cert_{b,j} = (m_{\omega b}, y_j, g_j)$.

BCA-VER$(1^{\lambda}, pk_{b,j}, Cert_{b,j}, pk_{CA}, pk_b) \rightarrow (\{1, \perp\})$*:* The sender a is required to verify the authenticity of public key $pk_{b,j}$ associated to receiver b's jth updated address $Adr_{b,j}$, in case the address has been replaced. The sender a takes the security parameter $1^{\lambda}, pk_b = (n_b, e_b), pk_{b,j}, Cert_{b,j}, pk_{CA}$ as input, and calculates

$$\begin{aligned} k'_j &\leftarrow g_j^{e_b} (\text{mod } n_b) \\ r'_j &\leftarrow y_j^{e_0} h(m_{\omega b}, n_b, e_b)^{k'_j} (\text{mod } n_0). \end{aligned} \tag{7}$$

The sender a checks whether $h(r'_j, n_{b,j}, e_{b,j}) = k'_j$ holds. If it holds, receiver b's jth updated wallet address $Adr_{b,j}$ is valid; otherwise, sender a terminates the protocol. The correctness of this verification lies in that

$$\begin{aligned} v_b^{e_0} &= h(m_{\omega b}, n_b, e_b)^{-1} (\text{mod } n_0) \\ k'_j &= g_j^{e_b} = k_j^{d_b e_b} (\text{mod } n_b) = k_j \\ r'_j &= t_j^{e_0} v_b^{k_j e_0} h(m_{\omega b}, n_b, e_b)^{k_j} = t_j^{e_0} = r_j (\text{mod } n_0). \end{aligned} \tag{8}$$

Therefore, we have

$$h(r'_j, n_{b,j}, e_{b,j}) = h(r_j, n_{b,j}, e_{b,j}) = k_j = k'_j. \tag{9}$$

C. LBS Message's Encryption and Decryption

MSG-ENC $(1^{\lambda}, m, pk_{b,j}) \rightarrow (c_m)$*:* In conventional blockchain, all stored valid messages are publicly available; while in our blockchain enforced VSNs, the LBS messages should be kept secret to anyone except the authorized receiver. The encryption process performs the following operations: the sender a takes LBS message $m \in \{0, 1\}^{\lambda_1}, pk_{b,j} = (n_{b,j}, e_{b,j})$ as input, randomly selects $r_j \in_R \{0, 1\}^{\lambda_0}$, and calculates

$$\begin{aligned} \omega_j &\leftarrow h_3(m \| r_j), \quad s_j \leftarrow g\big(\omega_j\big) \oplus \big(m \| r_j\big) \\ y_j &\leftarrow \big(\omega_j \| s_j\big), \quad c_m = y_j^{e_{b,j}} \text{mod } n_{b,j} \end{aligned} \tag{10}$$

where $\|$ denotes the concatenation operation, $h_3(\cdot), g(\cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^{\lambda}$ are cryptographic hash functions satisfying $\lambda = \lambda_0 + \lambda_1$, and c_m represents sender a's encryption on message m under receiver b's public key $pk_{b,j}$.

CPT-DEC $(1^{\lambda}, c_m, sk_{b,j}) \rightarrow (m)$*:* The decryption process is carried out by receiver b after receiving the encrypted LBS message c_m from the RSU. Taking $1^{\lambda}, c_m, (sk_{b,j} = d_{b,j})$ as input, it first verifies the authenticity of public key $pk_{a,j'}$ associated to sender a's j'th updated wallet address $Adr_{a,j'}$ using the algorithm in Section VI-B. If a's wallet address is invalid, the message would be discarded; otherwise, the RSU continues the following decryption process. It calculates $y_j = c_m^{d_{b,j}} (\text{mod } n_{b,j})$, parses y_j as $\omega_j \| s_j$, computes $g(\omega_j) \oplus s_j$ as $m \| r_j$, then it gets m by removing the last λ_0 bits of $m \| r_j$. Verifier compute $w_j = h_3(m \| r_j)$ to check the validity of the ciphertext c_m.

D. LBS Ciphertext's Signature Generation and Verification

MSG-SIG$(1^{\lambda}, c_m, pk_{a,j'}, sk_{a,j'}) \rightarrow (\sigma_{c_m})$*:* When uploading an LBS message, sender a is also required to generate a signature on the encrypted LBS message c_m generated by the algorithm MSG-ENC in Section VI-C for authentication. Assuming that sender a adopts its j'th updated wallet address, it takes the security parameter 1^{λ}, the ciphertext c_m, the public-key and secret-key pair $pk_{a,j'} = (n_{a,j'}, e_{a,j'}), sk_{a,j'} = (d_{a,j'})$ as input, randomly selects $r_{j'} \in_R \{0, 1\}^{\lambda_0}$, and calculates

$$\begin{aligned} \omega_{j'} &\leftarrow h_3\big(H(c_m) || r_{j'}\big), \quad s_{j'} \leftarrow g\big(\omega_{j'}\big) \oplus \big(H(c_m) || r_{j'}\big) \\ y_{j'} &\leftarrow \big(\omega_{j'} || s_{j'}\big), \quad \sigma_{c_m} = y_{j'}^{d_{a,j'}} \text{mod } n_{a,j'} \end{aligned} \tag{11}$$

where $\|$ denotes the concatenation operation, $h_3(\cdot), g(\cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^{\lambda}, H(\cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^{\lambda_1}$ are cryptographic hash functions satisfying $\lambda = \lambda_0 + \lambda_1$, and σ_{c_m} represents sender a's signature on ciphertext c_m. After the ciphertext of LBS message c_m and the associated signature σ_{c_m} are generated, sender a writes down the LBS communication data structure with the format demonstrated in Table I, and sends it to nearest RSU.

SIG-VER$(1^{\lambda}, \sigma c_m, c_m, pk_{a,j'}, Cert_{a,j'}, pk_{CA}) \rightarrow (\{1, \perp\})$*:* An encrypted LBS message will be logged into blockchain and sent to receiver b if and only if it has been successfully verified by the RSU. The RSU takes the security parameter $1^{\lambda}, \sigma_{c_m}, c_m, (pk_{a,j'}, Cert_{a,j'}, pk_{CA})$ as input, it first verifies the authenticity of public key $pk_{a,j'}$ associated to sender a's j'th updated wallet address $Adr_{a,j'}$ using the algorithm in Section VI-B. If a's wallet address is invalid, the message would be discarded; otherwise, the RSU continues the following verification process. It calculates $y_{j'} = \sigma_{c_m}^{e_{a,j'}} (\text{mod } n_{a,j'})$, parses $y_{j'}$ as $\omega_{j'}||s_{j'}$, computes $g(\omega_{j'}) \oplus s_{j'}$ as $H(c_m)||r_{j'}$, and checks whether $h_3(H(c_m)||r_{j'}) = \omega_{j'}$ holds. If it holds, the RSU logs the encrypted LBS message into the blockchain; otherwise, it is invalid and discarded. Note that the signature verification can be carried out by the nearest RSU node in the blockchain enforced VSNs. Finally, receiver b can also check the validity of signature by performing the same operations presented above as the RSU executes. If it fails, the encrypted LBS message is invalid; otherwise, b decrypts m by executing the algorithm CPT-DEC in Section VI-C.

E. Misbehavior Tracing

$MIS - TRA(pk_{b,j}, Cert_{b,j}) \rightarrow b$: To achieve conditional identity privacy preserving: vehicular user's identity privacy is well protected unless some misbehavior is detected. Specifically, if a short-term public key $pk_{b,j}$ is found to be suspicious, RSU can check the $m_{\omega b}$ in $Cert_{b,j}$, and refer to the relationship pair $(b, pk_b, m_{\omega b})$ stored in the account pool, the true identity b of the short-term public key $pk_{b,j}$ would be effectively traced.

VII. Security Analysis and Performance Evaluation

In this section, we present the security analysis and performance evaluation to demonstrate the effectiveness and efficiency of our proposed lightweight threshold CA LTCA for consortium blockchain and privacy-preserving LBS protocol PPVC in blockchain enforced VSNs.

A. Security Analysis

1) *Conditional Identity Privacy Preserving:* In our scheme, each vehicle u has a public key pk_u, and its jth updated wallet address $Adr_{u,j}$. Whenever a trade is executed, the other vehicle would only know that this message is from/to the address $Adr_{u,j}$ which is certificated by pk_u without learning anything about u's true identity. It is important to note that $Adr_{u,j}$ is also updated periodically to protect u's identity privacy against the background analysis attack. The conditionally identity privacy is achieved in the algorithm $MIS - TRA$ that an ordinary vehicle can only trace the public key pk_u by publicly verifying its certificate $Cert_{u,j}$; while only the RSU can find out the true identity u when a misbehavior is detected from its privately kept relationship pair (u, pk_u).

2) *Man-in-the-Middle-Attack Resistance:* It is noted that if the certificate of $pk_{b,j}$ associated to receiver b's jth updated wallet address $Adr_{b,j}$ is unforgeable, the binding relationship between receiver b and $Adr_{b,j}$ is guaranteed since $Adr_{b,j} = h'(n_{b,j})$ where $pk_{b,j} = (n_{b,j}, e_{b,j})$. It is observed that the certificate generation and verification is based on a newly designed proxy-protected proxy signature where receiver b generates the certificate $Cert_{b,j}$ with the proxy signing key v_b and her/his own secret key sk_b.

Theorem 1: The proposed Π_{Cert} is EUF-CMA secure, if GQ signature is EUF-CMA and hash function $h(\cdot)$ is collision resistant.

Proof: If the EUF-CMA security of Π_{Cert} is corrupted, then either the unforgeability of GQ scheme [29] or the collision resistance of $h(\cdot)$ would be broken.

1) If a collision can be found in the hash function $h(\cdot)$, then an adversary can effectively find a collision $h(m_{\omega u}, n_b, e_b) = h(m_{\omega \mathcal{A}_1}, n_{\mathcal{A}_1}, e_{\mathcal{A}_1})$ ($m_{\omega \mathcal{A}_1}$ includes the pseudonym of adversary $\mathcal{A}_1$), which would lead to the successful verification of $(pk_{\mathcal{A}_1}, Cert_{b,j})$ in (6) and (7).

2) If GQ signature can be forged, then in our scheme, an adversary can forge a certificate $Cert_{b,j}$ in (6) without any knowledge of v_b, since GQ signature [29] is constructed upon the following knowledge that $B^v \cdot J$ mod $n = 1$; while in the certificate generation of our scheme, we have $v_b^{e_0} h(m_{\omega b}, n_b, e_b) = 1$ mod n_0 where the parameters B, v, J in GQ signature [29] are, respectively, replaced by $v_b, e_0, h(m_{\omega b}, n_b, e_b)$ instead.

3) A coalition of corrupted RSUs S^{Cpt} ($|S^{Cpt}| < k$) cannot generate the authentic proxy signing key due to the unconditional security of Shamir's secret sharing. Therefore, it would not give any additional advantage for the adversary to forge the certificate of a wallet address.

Therefore

$$Adv_{\mathcal{A}_1, \Pi_{Cert}}^{\text{EUF-CMA}}\left(1^\lambda, t, q_c\right) \leq \frac{1}{2^\lambda} + \epsilon_{GQ}$$

where λ and ϵ_{GQ} are the security parameter and the probability to successfully forge a signature in the GQ's signature scheme. ∎

3) *LBS Message's Signature Unforgeability:* It is observed from the algorithm $MSG - SIG$ in Section VI-D that without sender a's secret key $sk_{a,j'}$ associated to her/his j'th updated wallet address $Adr_{a,j'}$, no adversary can forge the signature on LBS massage m. From (11), the EUF-CMA security of σ_{c_m} can be reduced to the RSA assumption and the collision resistance of cryptographic hash functions.

Theorem 2: The proposed Π_{Sig} is EUF-CMA secure, if RSA assumption holds and hash function $H(\cdot)$ is collision resistant.

Proof: If the EUF-CMA security of Π_{Sig} is corrupted, then either the RSA assumption or the collision resistance of $H(\cdot)$ would be broken.

1) If adversary $\mathcal{A}_2$ can find a collision in $H(\cdot)$ such that $H(c_m) = H(c'_m)$, then the forged signature (c'_m, σ_{c_m}) from (11) can pass the verification.

2) If the collision is not found but the adversary can still forge a valid signature $\sigma_{c'_m}$ on ciphertext c'_m, then the challenger can compute y'_j using the algorithm in (11), with the result that $\sigma_{c'_m}^{e_{a,j'}} = y'_j$ which solves the RSA problem.

Therefore

$$Adv_{\mathcal{A}_2, \Pi_{Sig}}^{\text{EUF-CMA}}\left(1^\lambda, t, q_c\right) \leq \frac{1}{2^\lambda} + \epsilon_{\text{RSA}}$$

where λ and ϵ_{RSA} are the security parameter and the probability to successfully solve the RSA problem. ∎

4) *LBS Message Confidentiality:* It is observed from the algorithm $MSG - ENC$ that the LBS message m is encrypted in (10) of Section VI-C, which means that the confidentiality of the LBS message can be reduced to the RSA assumption.

Theorem 3: The proposed Π_{Enc} is indistinguishably secure under CCA-attack, if RSA assumption holds.

Proof: The ind-cca-atk security of Π_{Enc} can be reduced to the RSA assumption for the following reasons.

1) If the RSA assumption can be solved by adversary $\mathcal{A}_3$, it can compute y_j without the knowledge of $d_{b,j}$, thus it can directly know the plaintext m from (10), from which the security of Π_{Enc} is apparently defeated.
2) On the other hand, for the indistinguishability under CCA-attack, our encryption has a validation process to check the validity by computing $w_j \leftarrow h_3(m\|r_j)$. Since the RSA assumption is hard to adversary $\mathcal{A}_3$, it cannot get the knowledge of y_j, hence w_j, which would disable a successful decryption query of a legal ciphertext. Hence, the decryption oracle is inaccessible to adversary $\mathcal{A}_3$.

Therefore

$$Adv_{\mathcal{A}_3,\Pi_{Enc}}^{ind-cca-atk}\left(1^{\lambda}, t, q_c\right) \leq \epsilon_{\text{RSA}}$$

where ϵ_{RSA} is the probability to successfully solve the RSA problem. ■

5) *CA's Misbehavior Detectability:* It is observed that CA's (RSU's) misbehavior can be detected in our scheme. The reason is that in the algorithm $PSK - REC$, once user u gets the proxy signing key v_u from RSUs, she/he can verify the authenticity of the key by checking whether $v_u^{e_0} h(m_{\omega u}, n_u, e_u) \equiv 1(\bmod\ n_0)$ holds. If it does not hold, misbehavior of CA (RSU) would be immediately detected.
6) *Single-Point Failure Resistance:* To prevent single-point failure and to split the trust, an architecture of threshold CA is proposed in our scheme. In the algorithm $CAK - DST$, each RSU plays the role of a partial CA by obtaining a partial secret key $d_i = f(i)$, which is exploited in the algorithm $PSK - GEN$ to generate the proxy signing key v_u for user u by the collaboration of no less than threshold number of RSUs.

B. Performance Evaluation

We study the performance and the efficiency comparison of our proposed lightweight threshold CA LTCA for consortium blockchain and privacy-preserving LBS protocol PPVC in blockchain enforced VSNs in the aspects of computational overhead, communication overhead, average delay, and delivery ratio.

Theoretical Analysis: Let l be the number of blockchain addresses of each vehicle. In Table III, we compare our proposed LTCA with Sharma *et al.*'s [13] blockchain-VN scheme which adopts a traditional CA [19], [30] (centralized CA). It is observed that both CA's computational and user's communication complexity of traditional CA [19] is $O(l)$ while our LTCA is $O(1)$, independent to the number of blockchain addresses authenticated. The reason is that in our scheme, once the threshold number of RSUs (CAs) authorize a legitimate vehicle by issuing a proxy signing key v_u at the offline phase only once in the algorithms $PSK - GEN$ and $PSK - REC$, the vehicle can authenticate all its l updated blockchain addresses itself without the intervention of the RSUs at the online phase in the algorithm $CRT - GEN$. However, traditional CA [19] requires a complicated authentication process every time a new blockchain address is updated and the vehicle is also required to communicate with CA several times each time it wants to authenticate a new blockchain address.

TABLE III
EFFICIENCY COMPARISON OF OUR PROPOSED LTCA

	CA's comp.	User's comm.
Our Proposed LTCA	$O(1)$	$O(1)$
Traditional CA [13], [19], [30]	$O(l)$	$O(l)$

TABLE IV
EFFICIENCY COMPARISON OF OUR PROPOSED PPVC

	Comp.	Comm.
Our PPVC	$O(1) \cdot P$	$O(1)$
RingCT 2.0[18]	$O(mn) \cdot P + O(mn^2 + t) \cdot M$	$O(m)$
RingCT[20]	$O(mn) \cdot M$	$O(mn)$

We also compare the computational cost and communication cost between RingCT [20], RingCT 2.0 [18] to achieve user's identity privacy exploiting the technique of ring signature and our proposed PPVC adopting periodically updating wallet addresses. Let n, m, and t, respectively, be the number of users in a group, the number of input accounts, and the number of output accounts. Table IV demonstrates the efficiency comparison where the symbols denote the following.

1) *M:* Cost of multiplication operation.
2) *P:* Cost of modular exponentiation operation.

For computational cost, it is observed that each user in RingCT [20] is required to execute $O(mn)$ modular exponentiation operations and $O(mn^2 + t)$ multiplication operations and each user in RingCT 2.0 [18] needs to perform $O(mn)$ multiplications; while in our PPVC, the computational complexity of modular exponentiation is $O(1)$, independent to the number of group members n, the number of input accounts m and output accounts t. It is also noted that the communication complexity of RingCT [20] and RingCT 2.0 [18] is, respectively, $O(m)$ and $O(mn)$; while in our PPVC, the communication complexity is $O(1)$, also independent to the parameters n, m, t. The reason is that our PPVC achieves user's identity privacy by devising a proxy-signature-based lightweight CA to periodically update the user's address with a low cost, and keeps a table of the binding between user's identity and their public key in RSU's account pool. On the contrary, RingCT [20] and RingCT 2.0 [18] achieve the same security goal by using either ring signature or linkable ring signature scheme, in which an anonymous group of n users is required to interact for cooperation on their massage signatures so that the adversary cannot distinguish the indeed user who actually signs the message in this group.

Experimental Results: We study the performance among ten vehicles and a total number of 50 RSUs in our blockchain

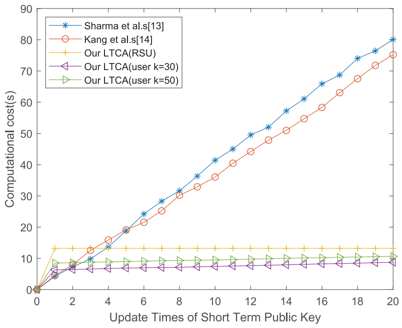


Fig. 4. Computational cost comparison of our proposed LTCA.

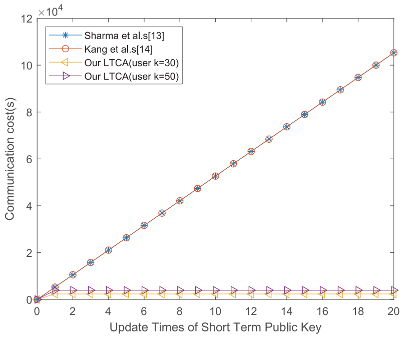


Fig. 5. Communication cost comparison of our proposed LTCA.

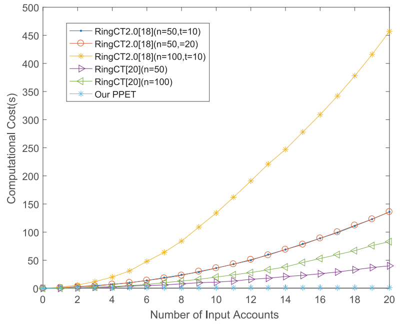


Fig. 6. Computational cost comparison of our proposed PPVC.

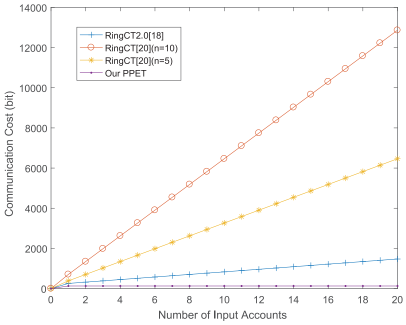


Fig. 7. Communication cost comparison of our proposed PPVC.

enforced VSNs by exploiting MIRACL library [31] running on Linux platform with 2.8-GHz processor. For trustable result, the experiment is based on a real transaction dataset in Blockchain [32], we consider the whole transaction as a message to sign in the signature scheme. Fig. 4 demonstrates the RSU's computational cost comparison in the process of certifying 20 short-term public keys of each vehicle. It is observed that CA's (RSU's) computational cost in our LTCA is approximately constant, while the computational cost of traditional CA [13], [19], [30] dramatically increases as the number of account addresses (i.e., update times of short-term public keys) increases. In our scheme, RSU is only needed to calculate a proxy-signing-key for vehicle once and for all. The future authenticate process does not need any more intervention from RSU. Therefore, the computation cost is nearly constant. However, in the traditional certificate scheme, CA needs to join the authenticate process each time a blockchain address is updated. Fig. 5 shows the user's communication cost of our LTCA is also significantly reduced compared to the traditional CA [19]. The reason is that in our LTCA user is only required to ask for the proxy-key in the proxy-key generation process. The user can authenticate her/his periodically updated wallet addresses by her/himself without the intervention of online RSUs (CAs). Finally, it is observed from both Figs. 4 and 5 that a single user's computational cost and communication cost slightly increase as the threshold number of RSUs increases from $k = 30$ to 50 to achieve a higher security level, since the increased complexity from a large threshold is only concerned with the calculation of a proxy-signing-key in algorithm $PSK - REC$ (i.e., independent to the update times of short-term public keys), which would be amortized to all times of updates.

Fig. 6 shows the user's computational cost of our PPVC is dramatically reduced compared to RingCT [20] and RingCT 2.0 [18], as the numbers of input accounts m, group members n, and output accounts t, respectively, increase. Fig. 7 illustrates the user's communication cost of our PPVC is significantly lower than both RingCT [20] and RingCT 2.0 [18] as the numbers of input accounts m and group members n, respectively, increase. The efficiency advantage results from the fact that a set of group members in both RingCT [20] and RingCT 2.0 [18] is required to interact for collaboration on generating the message signature in an identity privacy-preserving manner; while in our PPVC, it can be completed by the user her/himself by exploiting a lightweight CA to efficiently bind the user's true identity to their pseudonym and periodically updated account addresses.

Figs. 8 and 9, respectively, demonstrate the comparison of the average delay and delivery ratio between our proposed PPVC and [12] where an LBS system is implemented without blockchain, Sharma *et al.*'s [13] where a blockchain-based VANET is constructed by considering vehicles as miners and Kang *et al.*'s [14] where RSUs are adopted as miners under traditional CAs. It is obviously observed that in Fig. 8, the average delay in [12] is the most significant since the LBS messages are transmitted by vehicles in a peer-to-peer manner, where the high mobility and the intermittent of routing hamper the realtime LBS message delivery. Then, Sharma *et al.* [13] reached the average delay higher than [14] but lower than [12], by adopting the blockchain technique where resource-constrained vehicles serve as miners, which becomes a heel dragging factor since only little amount of computational resources are remained for actual LBS message

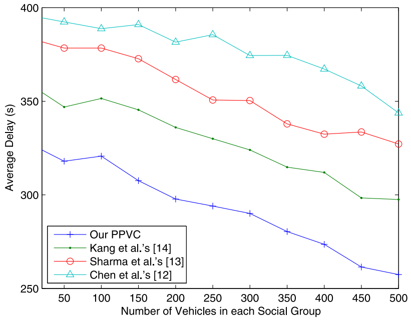


Fig. 8. Average delay comparison of our proposed PPVC.

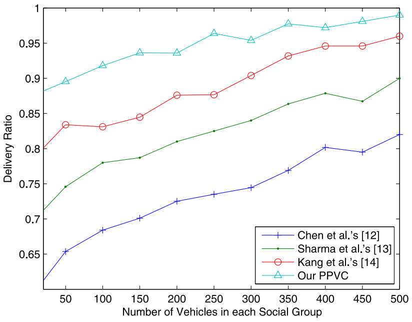


Fig. 9. Delivery ratio comparison of our proposed PPVC.

delivery; while in Kang *et al.*'s [14], the RSUs with more computational power and storage space are considered as miners. Finally, it is noted that our proposed PPVC reaches the lowest average delay. The reason is that Kang *et al.*'s scheme [14] still needs the vehicles to interact with an online CA to generate the corresponding public-key certificate for each updated blockchain address; while in our PPVC, the authorization from RSUs as decentralized CAs is required only once in the offline phase, and the certificates can be generated using the proxy signing key by the vehicles themselves without the intervene of an online CA where a large amount of communication overhead would be reduced. For the same reason, the delivery ratio of our proposed PPVC is also the most significant, which is presented in Fig. 9.

VIII. Conclusion

In this article, a lightweight threshold CA LTCA for consortium blockchain and an efficient privacy-preserving LBS protocol PPVC in blockchain enforced VSNs are proposed. In our approach, based on a newly devised threshold proxy signature, a coalition of a threshold number of CAs (RSUs) is required to generate a proxy signing key only once for each legitimate user in the offline phase. This allows vehicles to authenticate themselves every time an updated account address is generated in the online phase without involving an online CA. We demonstrated both the security and performance efficiency of the proposed approach. Future improvement can be made by scattering the account pool so that no single RSU could expose the user's true identity and a better consensus protocol for consortium blockchain might be found and applied to our proposal.

References

[1] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular social networks: Enabling smart mobility," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 49–55, May 2017.

[2] H. Wang, L. Wan, M. Dong, K. Ota, and X. Wang, "Assistant vehicle localization based on three collaborative base stations via SBL-based robust DOA estimation," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5766–5777, Jun. 2019.

[3] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, "Efficient and privacy-preserving truth discovery in mobile crowd sensing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3854–3865, Apr. 2019.

[4] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 146–152, Jun. 2017.

[5] S. Rowan, M. Clear, M. Gerla, M. Huggard, and C. Goldrick, "Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels," 2017. [Online]. Available: arXiv.1704.25.53.

[6] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, and X. Lin, "PTAS: Privacy-preserving thin-client authentication scheme in blockchain-based PKI," *Future Gener. Comput. Syst.*, vol. 96, pp. 185–195, Jul. 2019.

[7] Z. Zhou, B. Wang, M. Dong, and K. Ota, "Secure and efficient vehicle-to-grid energy trading in cyber physical systems: Integration of blockchain and edge computing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 43–57, Jan. 2020.

[8] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong, "Location privacy in usage-based automotive insurance: Attacks and countermeasures," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 1, pp. 196–211, Jan. 2019.

[9] M. Mihaylov, S. Jurado, N. Avellana, K. Van Moffaert, I. De Abril, and A. Nowe, "NRGcoin: Virtual currency for trading of renewable energy in smart grids," in *Proc. 11th Int. Conf. Eur. Energy Market (EEM)*, 2014, pp. 1–6.

[10] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Depend. Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep./Oct. 2018.

[11] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 17–30.

[12] Y.-S. Chen, T.-T. Lo, C.-H. Lee, and A.-C. Pang, "Efficient pseudonym changing schemes for location privacy protection in VANETs," in *Proc. Int. Conf. Connected Veh. Expo (ICCVE)*, 2013, pp. 937–938.

[13] P. K. Sharma, S. Moon, and J. Park, "Block-VN: A distributed blockchain based vehicular network architecture in smart city," *J. Inf. Process. Syst.*, vol. 13, no. 1, pp. 184–195, 2017.

[14] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[15] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using P2P network traffic," in *Proc. Financ. Cryptography Data Security*, 2014, pp. 469–485.

[16] A. Zhang and X. Ma, "Decentralized digital certificate revocation system based on blockchain," *J. Phys. Conf. Series*, vol. 1069, Jun. 2018, Art. no. 012125.

[17] Y. Zhang, C. Xu, J. Ni, H. Li, and X. S. Shen, "Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage," *IEEE Trans. Cloud Comput.*, early access, doi: 10.1109/TCC.2019.2923222.

[18] S. Sun, M. Au, J. Liu, and T. Yuen, "RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero," in *Proc. Eur. Symp. Res. Comput. Security*, 2017, pp. 456–474.

[19] A. Yakubov, W. M. Shbair, A. Wallbom, and D. Sanda, "A blockchain-based PKI management framework," in *Proc. IEEE/IFIP NOMS Int. Workshop Manag. Blockchain (Man2Block)*, 2018, pp. 1–6.

[20] S. Noether. (2015). *Ring Signature Confidential Transactions for Monero*. [Online]. Available: http://eprint.iacr.org/

[21] L. Abrams. (2018). *Clipboard Hijacker Malware Monitors 2.3 Million Bitcoin Addresses*. Accessed: Jul. 11, 2018. [Online]. Available: https://www.bleepingcomputer.com/news/security/clipboard-hijacker- malware-monitors-23-million-bitcoin-addresses/

[22] G. Zyskind and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Security Privacy Workshops (SPW)*, San Jose, CA, USA, 2015, pp. 180–184.

[23] L. Li *et al.*, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018.

[24] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[25] J.-C. Cheng, N.-Y. Lee, C. Chi, and Y.-H. Chen, "Blockchain and smart contract for digital certificate," in *Proc. IEEE Int. Conf. Appl. Syst. Invention (ICASI)*, 2018, pp. 1046–1051.

[26] J. Chen, S. Yao, Q. Yuan, K. He, S. Ji, and R. Du, "CertChain: Public and efficient certificate audit based on blockchain for TLS connections," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, 2018, pp. 2060–2068.

[27] W. Feng and Z. Yan, "MCS-Chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain," *Future Gener. Comput. Syst.*, vol. 95, pp. 649–666, Jun. 2019.

[28] L. G. Guillou and J.-J. Quisquater, "A 'paradoxical' identity-based signature scheme resulting from zero-knowledge," in *Proc. Adv. Cryptol. (CRYPTO)*, 1990, pp. 216–231.

[29] (2018). *X.509: Information Technology—Open Systems Interconnection—The Directory: Public-Key and Attribute Certificate Frameworks*. Accessed: Oct. 10, 2018. [Online]. Available: http://www.itu.int/rec/T-REC-X.509/en

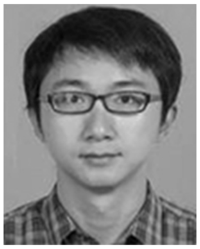
[30] *Multiprecision Integer and Rational Arithmetic C/C++ Library*. Accessed: Oct. 10, 2018. [Online]. Available: http://www.shamus.ie

[31] (2018). *Bitcoin Block Explorer—Blockchain*. Accessed: Aug. 31, 2018. [Online]. Available: https://www.blockchain.com/explorer

[32] A. Kiayias, A. Russell, B. David, and R. Oliynykov. *Ouroboros: Aprovably Secure Proof-of-Stake Blockchain Protocol*. Accessed: Feb. 23, 2019. [Online]. Available: http://eprint.iacr.org/2016/889

Huajie Shen is currently pursuing the master's degree with the Department of Cryptography and Network Security, East China Normal University, Shanghai, China.

His research interests mainly include blockchain security, cryptocurrency protocols, and privacy preserving in VANETs.



Jun Zhou received the Ph.D. degree in computer science from the Trusted Digital Technology Laboratory, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

He is currently an Associate Professor with the Department of Cryptography and Network Security, East China Normal University, Shanghai. His research interests mainly include secure multiparty computation, AI security and blockchain privacy-preserving in cryptography, and information security.

Zhenfu Cao (Senior Member, IEEE) received the B.Sc. degree in computer science and technology and the Ph.D. degree in mathematics from the Harbin Institute of Technology, Harbin, China, in 1983 and 1999, respectively.

He was exceptionally promoted to an Associate Professor in 1987, became a Professor in 1991, and is currently a Distinguished Professor with East China Normal University, Shanghai, China. Since 1981, over 400 academic papers have been published in journals or conferences. He is the Leader of the Asia 3 Foresight Program (61161140320), and the Key Project (61033014, 61632012) of the National Natural Science Foundation of China. His research interests mainly include number theory, cryptography, and information security.

Dr. Cao has received a number of awards, including the Youth Research Fund Award of the Chinese Academy of Science in 1986, the Ying-Tung Fok Young Teacher Award in 1989, the National Outstanding Youth Fund of China in 2002, the Special Allowance by the State Council in 2005, the First Prize of Natural Science of Ministry of Education in 2018, and the 2007 IEEE International Conference on Communications-Computer and Communications Security Symposium Best Paper Award in 2007. He serves as a member of the Expert Panel of the National Nature Science Fund of China.

Xiaolei Dong received the Doctoral degree from the Harbin Institute of Technology, Harbin, China, in 2001.

She was a Postdoctoral Fellow with Shanghai Jiao Tong University, Shanghai, China, from September 2001 to July 2003, where she joined the Department of Computer Science and Engineering, in 2003. In 2014, she joined East China Normal University, Shanghai, where she is currently a Distinguished Professor. She hosts a number of research projects supported by the National Basic Research Program of China (973 Program), and the special funds on information security of the National Development and Reform Commission and the National Natural Science Foundation of China. Her primary research interests include number theory, cryptography, and trusted computing.

Dr. Dong received the First Prize of the China University Science and Technology Award in 2002 for her Number Theory and Modern Cryptographic Algorithms project. Her New Theory of Cryptography and Some Basic Problems project received the Second Prize of the Shanghai Nature Science Award in 2007. Her Formal Security Theory of Complex Cryptographic System and Applications project received the Second Prize of the Ministry of Education Natural Science Progress Award in 2008.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. degree in information security from Queensland University of Technology, Brisbane, QLD, Australia, in 2006.

He currently holds the Cloud Technology Endowed Professorship with the University of Texas at San Antonio, San Antonio, TX, USA.

Dr. Choo was a recipient of various awards, including the ESORICS 2015 Best Paper Award, the Winning Team of the Germany's University of Erlangen–Nuremberg (FAU) Digital Forensics Research Challenge 2015, and the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency. He is an Australian Computer Society Fellow.