Secure and Efficient Blockchain-Assisted Authentication for Edge-Integrated Internet-of-Vehicles

Meng Shen, *Member, IEEE*, Hao Lu, Fei Wang, *Member, IEEE*, Huisen Liu, and Liehuang Zhu, *Senior Member, IEEE*

***Abstract*—Edge-Integrated Internet-of-Vehicles (IoV) sinks service to edge nodes, which responses quickly to vehicles' requests and alleviates the burden of cloud servers. In such scenario, the entities of IoV need to be mutually authenticated since potential attacks can impersonate edge nodes to send false instructions to vehicles, or impersonate legitimate service subscribers to get free-ride IoV services. Furthermore, due to the high mobility characteristics of vehicles, frequent authentication is required between vehicles and edge nodes. We hope that the authentication process can be conducted efficiently to ensure the continuous service. Existing works cannot balance the security and efficiency well. This is still an issue worthy of discussion. In this article, we propose SEA, a secure and efficient blockchain-assisted authentication scheme for IoV. SEA achieves mutual authentication among vehicles, edge nodes and cloud servers. Specifically, the cloud server is only involved when vehicles are initially authenticated. And edge nodes realize the authentication of vehicles by querying the authentication result recorded by cloud on the blockchain, which significantly reduces the cryptographic computation overhead and eliminates network communication delay. Besides, session keys between any two entities involved are negotiated, which can secure sensitive data of vehicles. Extensive experiments have been conducted to show the security and efficiency of SEA.**

***Index Terms*—Blockchain, identity authentication, Internet-of-Vehicles.**

I. Introduction

INTERNET-of-Vehicles (IoV) [1], [2] is deemed as one of the most promising paradigms of Internet-of-Things (IoT) [3]. In IoV, vehicles can get access to a wide range of delay-sensitive and location-aware services, such as road traffic monitoring [4], emergency events alarming [5] and cloud-assisted autonomous driving [6], [7].

Manuscript received 27 February 2022; revised 4 May 2022; accepted 4 July 2022. Date of publication 27 July 2022; date of current version 14 November 2022. This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1006101, in part by Beijing Nova Program under Grant Z201100006820006, in part by NSFC Projects under Grants 61972039, 61902376, and 61872041. The review of this article was coordinated by Dr. Zehui Xiong. *(Corresponding author: Meng Shen.)*

Meng Shen and Liehuang Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenmeng@bit.edu.cn; liehuangz@bit.edu.cn).

Hao Lu and Huisen Liu are with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: luhao1999@bit.edu.cn; liuhuisen@bit.edu.cn).

Fei Wang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangfei@ict.ac.cn).

Digital Object Identifier 10.1109/TVT.2022.3194008

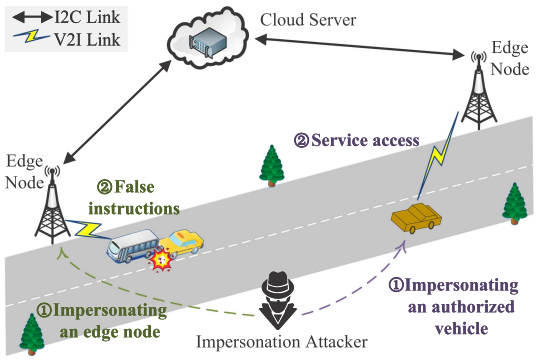

Fig. 1. A typical scenario illustrating potential impersonation attacks in edge-integrated IoV.

Different from IoT paradigms [8], IoV is a mobile network with dynamic topology and the vehicles within the network should remain connection with cloud servers while moving arbitrarily. In order to achieve this, two basic communication are defined: V2I (vehicles to infrastructure) and I2C (infrastructure to cloud). The edge node (e.g., the base station or roadside unit) is a typical infrastructure which takes charge of the information flow in its own scope. Vehicles can achieve dynamic access to IoV services through the intermediary of different edge nodes. With the rapid development of mobile edge computing, some real-time services are expected to be deployed at the edge nodes to provide better user experience.

The sinking of service to edge and the mobile characteristic of the nodes in IoV determines that it is more susceptible to malicious attackers. Fig. 1 depicts a typical Edge-Integrated IoV scenario where a fast-moving vehicle leads to frequent service handovers from one edge node to another. Since the exchanged information transmitted among three entities can be captured, modified or forged, a potential adversary can either impersonate an edge node to send false instructions to misdirect vehicles [9], or deceive edge nodes to mimic an authorized vehicle to get services. Therefore, an authentication scheme needs to be crafted to solve these security problems before the IoV paradigm being widely deployed.

However, it is a challenging task to design a secure and efficient authentication scheme in edge-integrated IoV.

First, the authentication scheme should take account of both security and privacy. On the one hand, most authentication

schemes in IoT [10] only consider the authentication between devices and service providers, which ignore the essential role of edge nodes in Edge-Integrated IoV. Given it, a security authentication scheme must achieve bilateral authentication among three entities to defend against the man-in-the-middle attack [11], which may deliver false instructions performing terrorism activities or visit unsubscribed IoV services for fee waiver [12], [13], [14]. On the other hand, edge nodes are usually assumed as *honest*-but-*curious*, they may spy on sensitive information transferred between the vehicle and the cloud while providing network services. Besides, the information exchanged between three entities may also be obtained by external attackers. Thus, the authentication scheme should also contribute to protecting privacy data during subsequent interactions between entities.

Second, due to the high-mobility of vehicles, the authentication process should be performed efficiently to guarantee continuous services. In edge-integrated IoV, a vehicle usually moves quickly from one edge node to another, which means that authentication happens frequently between the vehicle and each edge node along its route. The authentication process with high communication delay is prone to cause interruption of IoV services. To ensure the continuity of service, the authentication process between vehicles and edge nodes is desired to be conducted in a time-efficient manner.

A series of existing studies attempt to provide authentication solutions to IoV applications. In *cloud*-based schemes [14], [15], [16], [17], [18], a vehicle is authenticated by a centralized cloud server, which suffers from high and uncertain network communication delay caused by long-distance data transmission (as evaluated in Section VII) [19]. In *proxy*-based schemes [20], [21], [22], [23], dynamic proxies are selected from nearby vehicles, which heavily depends on vehicle density scattered around the target vehicle. Most of the above schemes use conventional centralized way to manage user's authentication information which relies on a Trusted-Authority (TA). Such mode may cause problems since it has the risk of personal data manipulation and single point of failure [24]. An attack on TA could lead to a crash of the authentication system. Besides, in IoV, a large number of vehicle authentication requests arrive at the core network all the time. Centralized mode undoubtedly brings performance bottlenecks, which is intolerable for real-time IoV services.

Recently, due to the decentralized and tamper-proof characteristics, blockchain draws the attention of researchers and has been proved useful in edge computing [25]. As a distributed database, blockchain solves the demerits deriving from centralization and provides stronger security guarantee by cryptography. In *blockchain*-based schemes [26], [27], [28], blockchain technology is exploited as a trusted ledger to assist identity authentication. However, blockchain in these schemes is usually deployed on the authentication side to achieve credible trust management, which does not solve the inherent drawbacks of the above two types of schemes (network delay or instability) and not bring significant efficiency gains. How to combine the blockchain to design a more efficient authentication process to adapt to the high-speed vehicle movement is still a problem worth exploring.

In this paper, we propose a Secure and Efficient blockchain-assisted Authentication (**SEA**) scheme for edge-integrated IoV. We design the initial authentication and re-authentication processes, and achieve mutual authentication securely among vehicles, edge nodes and cloud servers in the two processes. Given interaction with distant cloud server may cause long time delay, specifically, we transform the re-authentication process into a query on blockchain ledger which shares authentication results endorsed by cloud server. This significantly improves efficiency. The tamper-proof consortium blockchain guarantees the integrity of the authentication results. And during the authentication process, independent session keys are negotiated between any two entities, which contributes to the achievement of privacy-preservation.

The main contributions of this paper are as follows:

- We propose SEA, a blockchain-assisted authentication scheme. With dedicated design on the interaction among three entities, it achieves secure mutual authentication.
- We design the time-efficient re-authentication process by exploiting consortium blockchain to share authentication results, which significantly reduces cryptographic computation overhead and eliminates the authentication latency introduced by communication with the cloud server.
- We provide rigorous security analysis to demonstrate the security and privacy of SEA and evaluate its performance by extensive experiments. The experimental results prove its effectiveness and efficiency.

We organize the remainder of this paper as follows. We first review the existing authentication schemes related to IoV in Section II. We also illustrate background knowledge in Section III. Then, we present an overview of the proposed scheme in Section IV and describe the design details in Section V. Next, we present security analysis in Section VI and conduct extensive experiments to demonstrate its effectiveness and efficiency in Section VII. Finally, we give further discussion in Section VIII and conclude this paper in Section IX.

II. Related Work

In this section, we review the existing authentication solutions in IoV. We roughly classify them into three categories.

A. Cloud-Based Authentication Schemes

Cloud servers plays an indispensable role in the authentication process of these schemes. Wang et al. [29] propose a privacy-preserving cloud-based road condition monitoring system with source authentication (RCoM). It ensures the legitimacy of vehicles by achieving the authentication between vehicles and roadside units. Jiang et al. [14] present a cloud-centric three factor authentication scheme, which exploits passwords, smart cards and biometrics to provide guarantee of secure access to both cloud and autonomous vehicles. An privacy protection authentication scheme oriented to the multi-cloud environment is proposed by Cui et al. [16]. Shen et al. [18] propose a batch verification scheme for secure real-time traffic data aggregation for vehicular cloud.

However, the involvement of cloud servers brings network communication latency [33], [34]. Due to the complexity and uncertainty of the Internet [35], the latency may fluctuate a lot, which is unendurable for latency-sensitive applications.

B. Proxy-Based Authentication Schemes

In these schemes, proxies are introduced to facilitate the authentication process, which can be edge nodes or dynamic vehicles. Zhang et al. [30] propose an edge computing-based privacy-preserving authentication scheme for vehicular networks. It selects specific vehicles as edge computing vehicles to authenticate other vehicles and exploits the base station to relay messages. Liu et al. [20] select proxy entities among vehicles and the results obtained from them can be independently verified by roadside units. To reduce the burden of traffic control center, Song et al. [21] divide IoV into several fogs, each of which acts as a proxy to authenticate vehicles. Sutrala et al. [23] design a conditional privacy-preserving authentication supporting batch verification for messages by cryptography technique. Both vehicles and RSUs can authenticate their neighboring vehicles in batch mode.

Although proxies reduce the burden that central node authenticates all vehicles, this pattern may be out of control due to the vehicles' high-speed mobility [36]. And secrets should be pre-distributed to proxies that cannot guarantee reliability.

C. Blockchain-Based Authentication Schemes

Blockchain is applied in authentication, due to its characteristics of decentralization and immutability. Sharma et al. [31] propose a blockchain based architecture for vehicular information system with seamless access control. But this scheme does not consider the potential security threats of edge nodes. On the basis of consensus algorithm, Wang et al. [26] present a decentralized authentication scheme for IoV to guarantee the safety of mobile services. However, the authentication process still requires the participation of the cloud, causing some communication delay. Ma et al. [27] propose a lightweight mutual authentication with distribute key management (DB-KMM) and key agreement based on the bivariate polynomial. However, this scheme only describes the mutual authentication between the vehicle and the RSU, which is obviously not applicable to the general scenario of the IoV. Gabay et al. [32] propose a blockchain-based authentication scheme adopting zero-knowledge proof to provide guarantee for privacy-preserving of sensitive information. Liu et al. [28] present a blockchain-assisted group-authentication scheme based on dynamic proxies. In this scheme, blockchain is adopted to implement trust management of proxy vehicles. But the scheme mainly focuses on the security guarantee and neglects the low overall efficiency that may be introduced by complex consensus algorithm.

In a nutshell, the existing blockchain-based authentication schemes can not balance the security and efficiency in IoV scenario. Our scheme solves the above problem and the privacy of the entities involved is also well preserved.

III. Preliminary Preparation

In this section, we illustrate background knowledge of cryptography algorithms involved in SEA. And as the core component of our scheme, the consortium blockchain will be introduced later.

A. Related Encryption Knowledge

1) Elliptic Curve Cryptography (ECC): ECC is an asymmetric encryption algorithm based on elliptic curve mathematical theory. Compared to RSA, it can provide the same level of security with smaller key size [37]. Therefore, we choose it as the basic encryption method in SEA for IoV. The algorithm is defined in a cyclic subgroup of elliptic curve discrete finite field, which can be uniquely described by a six-tuple (p, a, b, G, n, h), the parameters will be further explained in Table II. The key pair of ECC is generated using the following rules:

- *Private key:* A random number $q \in_R Z_p^*$, where Z_p^* indicates positive integers less than p.
- *Public key:* Given the base point G, the public key Q can be calculated by $Q = qG$.

It's a discrete logarithm problem that using Q and G to calculate q, ECC achieves encryption by this feature [38].

2) Elliptic Curve Diffie-Hellman Ephemeral (ECDHE): ECDHE is an anonymous key exchange protocol allowing two users to derive a shared key over an insecure channel. Two interacting entities $\mathcal{A}$ and $\mathcal{B}$ respectively generate their own key pairs $(q_{\mathcal{A}/\mathcal{B}}, Q_{\mathcal{A}/\mathcal{B}})$ on the same elliptic curve and exchange public keys. Since $sk = q_A Q_{\mathcal{B}} = q_{\mathcal{B}} Q_{\mathcal{A}}$, they finally get secret sk without the exposure of private key. Within a polynomial time, it is not feasible to learn the secret key $q_{\mathcal{A}} \cdot q_{\mathcal{B}} \cdot P$ by $Q_{\mathcal{A}}$ and $Q_{\mathcal{B}}$ [39].

3) Elliptic Curve Digital Signature Algorithm (ECDSA): ECDSA is a combination of ECC and digital signature algorithm (DSA), which can ensure the validity of public key and information integrity in SEA. The generation and verification of the signature follow the rules below:

- *Signature generating process:*
  Step 1: Generates key pairs (q, Q) where $Q = q \cdot G$.
  Step 2: Randomly choose an integer k. Then compute $K = kG$ and $r = x_K \bmod n$, where x_K refers to the horizontal coordinate of point K. If $r = 0$, return step 1.
  Step 3: Compute the hash value C of message m. s will be computed according to (1). If $s = 0$, return step 2, otherwise the signature (r, s) is successfully generated.
  $$s = k^{-1}(C + qr) \bmod n. \tag{1}$$
- *Signature verification process:*
  The verifier holds the signature (r, s), the sender's public key Q, message m and the same ECC parameters.
  Step 1: Compute the hash value C of message m.
  Step 2: Compute $w = s^{-1} \bmod n$.
  Step 3: Compute $u_1 = Cw \bmod n$ and $u_2 = rw \bmod n$.
  Step 4: Compute $X = u_1 G + u_2 Q$. Accept the signature if and only if $x_X \bmod n = r$, where x_X refers to the horizontal coordinate of point X.

B. Blockchain Preliminary

Blockchain [40] has been attempted to be applied to different sectors [41], [42] in addition to traditional finance [43]. It is a kind of composite technology, which is essentially a distributed appended-only ledger. Compared with public blockchain, consortium blockchain is a more efficient, practical blockchain

TABLE I
SUMMARY OF TYPICAL EXISTING AUTHENTICATION SCHEMES FOR IOV

	Research	Technology	Pros	Cons
cloud-based schemes	Wang et al. [12]	token distribution protocol	Privacy-preserving monitoring;	RA-dependent
	Jiang et al. [11]	Biometrics; Smart cards	Biometric privacy protected	Unidirectional authentication
	Cui et al. [13]	ECC	Conditional privacy protection	Restraint to multi-cloud environment
	Shen et al. [14]	Message recovery signature	Privacy preservation; Batch verification	Computation-extensive
proxy-based schemes	Zhang et al. [24]	ECC; Fuzzy Logic	Key Agreement; Batch Authentication	Lack of certification of edge nodes
	Liu et al. [15]	Digital signature	Batch verification	Message-oriented
	Song et al. [16]	ECC; Deep learning	Security monitoring	Identity-based; Privacy leakage
	Sutrala et al. [17]	ECC	Conditional privacy preserving	CA-dependent
blockchain-based schemes	Wang et al. [21]	PKI-based	Decentralization authentication	Key distribution-dependent
	Ma et al. [22]	Bivariate polynomial	Decentralized key management	CA-dependent
	Gabay et al. [25]	Zero knowledge proofs	Privacy-preserving	Token distribution-dependent
	Liu et al. [23]	Pseudo-random	Collaborative authentication	Aggregated server-dependent

TABLE II
NOTATIONS OF THE PROPOSED SEA SCHEME

Symbol	Description
p	Parameter of elliptic curve base field
a	Parameter of elliptic curve equation
b	Parameter of elliptic curve equation
G	The generator the subgroup of the elliptic curve
n	The order of subgroup
h	Cofactor related to n
N	The order of the curve
(s, P)	A private-public key pair
$str1 \| str2$	Concatenation of $str1$ and $str2$
SID	Service id for a vehicle
$Sig_{\mathcal{A}}(str)$	Signature on str signed by entity $\mathcal{A}$
$Ver_{\mathcal{A}}(sig)$	Verification on a signature performed by entity $\mathcal{A}$
SR	Service request
AR_{SID}	Authentication result to a vehicle for its service SID
$AR_{\mathcal{AB}}$	Authentication result to $\mathcal{B}$ authenticated by $\mathcal{A}$
TTL_{SID}	Survival time of AR_{SID}
$\mathcal{S}$	A cloud server
$\mathcal{E}$	An edge node
$\mathcal{V}$	A vehicle

technology with light-weight and faster consensus process controlled by pre-selected miners [44]. We concentrate on consortium blockchain, which is expected to make a difference in the field of identity authentication, since it occupies the following advantages:

- *Decentralization.* The blockchain ledger is maintained by a group of nodes without central or third party intervention. This functionality is achieved by the consensus mechanism operated by all the peer nodes, which is supported by its underlying peer-to-peer network.
- *Immutability.* Anyone including the peer nodes cannot modify the historical ledger without the majority's consent. Only authorized peer nodes can update the ledger under the endorsement of a group of proper peer nodes.
- *Reliability.* Each peer node duplicates the blockchain ledger on its local file system. Data recorded on the ledger will not be lost as long as one peer node remains a copy.
- *Channel.* Channel mechanism is a native design in the consortium blockchain. Different nodes can be divided into different channels, which logs their own ledger that is invisible to nodes outside their channel.
- *Chaincode.* Peer nodes can only append their channel ledger through chaincode. Each chaincode is authorized to particular channels, which means authorizing to a group of nodes.

IV. THE PROPOSED SEA

In this section, we present the system architecture and describe the threat model. Then, we elaborate on the design goals that the proposed scheme should achieve.

A. System Architecture

We envision that there are mainly four kinds of parts in the architecture of SEA, namely cloud servers, edge nodes, vehicles and blockchain networks as depicted in Fig. 2.

- *Cloud server*, which is managed by IoV service provider to offer various services to vehicles, such as accident detection [45]. It performs initial authentication when a vehicle first gets access to its services, which verifies the qualification of the vehicle. To promote authentication efficiency in a fast-moving scenario, the cloud server endorses the initial authentication result on a blockchain ledger as described below, from which the edge nodes along the vehicle's route can conduct quick re-authentication to ensure service continuity.
- *Edge node* is the entity near to vehicles, which brings services closer to the end users so as to reduce the communication latency. In 4G-LTE architecture [46], it can be composed of a base station and an edge server deployed by Internet service providers. With the help of virtualization technology, it can support several virtual edge servers simultaneously, where each edge server can be allocated to a service. In SEA, when a vehicle shifts from one edge node to another, the new edge node will perform the re-authentication process. Specifically, it just queries the authenticity of the vehicle from the blockchain ledger which records the authentication result, instead of communicating with the cloud server.

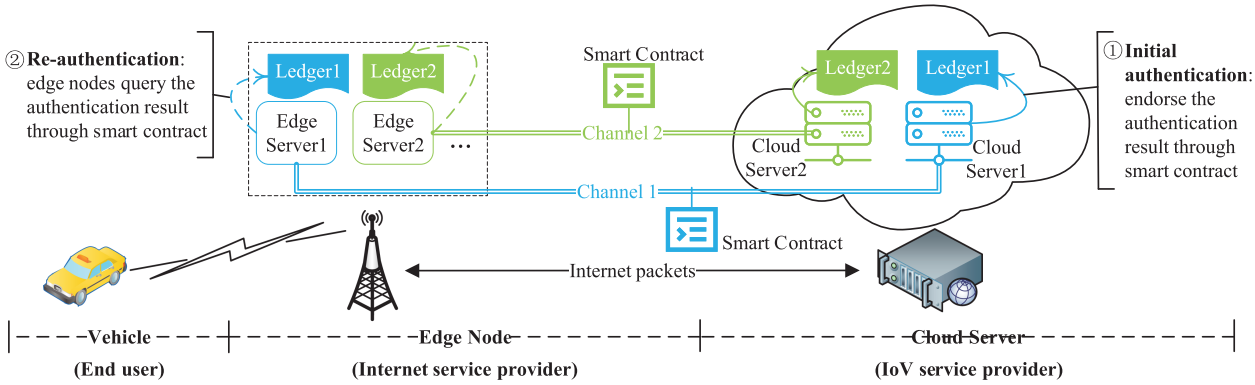

Fig. 2. Architecture overview of blockchain-assisted authentication scheme for edge-integrated IoV.

- *Vehicle:* A vehicle is a typical user terminal, which is also a platform for consumers to enjoy their subscribed IoV services. When a vehicle first requests services, it will be authenticated by the cloud server. Afterwards, every time the vehicle shifts from one edge node to another, it will be re-authenticated by the new edge node.
- *Blockchain Network* is essentially a virtual overlay peer-to-peer (P2P) network [47]. In our SEA, We introduce consortium blockchain as the fundamental component which is jointly maintained by edge nodes and the cloud server. And the blockchain service on edge node is supported by the Internet service provider, who can make revenue by providing computing resources close to users and storage resources that support fast read and write outputs [48], [49]. The edge node can be considered as peer node in the blockchain, which is both an endorser and a committer taking charge of the execution of smart contracts, block validation and the maintenance of the world state. Specially, an edge node may hold multiple ledgers and smart contracts, since each edge node groups for different services are supposed to share a corresponding independent ledger. The cloud server plays a supervisory role in the blockchain, who can identify admitted member nodes, define network and channel access authority. And it has the ability to update all blockchain ledgers.

In our scheme, the underlying blockchain shares the authentication result endorsed by cloud sever and provides a consensus service for the edge server deployed in edge node. The edge node can query and update the ledgers through the chaincode. Since the consortium blockchian endows peer nodes with trust [50], it prefers to use more efficient consensus algorithms such as PoS (Proof-of-Stake), PBFT (Practical Byzantine Fault Tolerance). It is noted that the blockchain is used as an auxiliary component in our SEA, so we don't declare the consensus mechanism of the consortium blockchain, which depends on the specific application scenario.

B. Threat Model

Following the common assumption on threat models in the existing literature [26], [27], [32], [51], we assume the threat model for the proposed scheme.

The cloud server is assumed to be honest, since it needs to provide better service for profits. And it attempts to identify authorized edge nodes and subscribers to avoid property loss.

The edge node is assumed to be honest-but-curious, since it is logically controlled by IoV service provider but physically owned by Internet service provider. It can fetch data from cloud servers. Thus, edge node has the ability and intention to peek sensitive data.

The adversary is a malicious user who tries to threaten the security and privacy of the IoV service. It can control only their own vehicles, but is assumed to have the ability to capture any packets transmitted over the network. In terms of security threats, he/she may impersonate any part of the three entities by replay attack or impersonation attack. For instance, the adversary may try to impersonate as legal vehicles to consume IoV services free of charge, or try to counterfeit as an edge node or a cloud server for delivering false instruction to vehicles. In terms of privacy threats, honest-but-curious edge nodes may obtain sensitive information while assisting the session between cloud servers and vehicles. External attackers may also attempt to obtain exchange information among three entities when IoV services are provided.

C. Design Goals

1) Performance Objectives: The proposed authentication scheme should enable efficient re-authentication between vehicles and edge nodes.

2) Security Objectives:

- *Mutual authentication*: To ensure the authenticity and legitimacy of the participants, vehicles and edge nodes as well as cloud servers should authenticate each other before data access happens.
- *Session key agreement*: A vehicle needs to negotiate an independent session key with every edge node. Similarly, each edge node needs to negotiate a distinct session key with the corresponding cloud server.
- *Privacy-preservation*: Personal sensitive data delivered through curious edge nodes should be well protected during the provision of IoV service.
- *Perfect forward security*: Assume that an adversary has cracked the current session key, he still cannot get the

previous session key through the captured messages. This property aims to protect the secrecy of previous communication.

V. Design Details of SEA

In this section, we elaborate the design details of SEA. Notations used in SEA are marked in Table II. Processes $Sig_{\mathcal{A}}(str)$ and $Ver_{\mathcal{A}}(sig)$ in the table both use ECDSA.

A. System Overview

SEA mainly includes three parts, namely system initialization, edge node/vehicle registration, authentication and key negotiation. Specially, the authentication phase are divided into two stages according to vehicles' authentication status: initial authentication and re-authentication. The cloud server is only involved in the initial authentication process when an unverified vehicle first access its service. Different cloud servers (which provide different Internet services) write authentication results into different ledgers related to their corresponding channel. When the vehicle switches between different edge nodes, the new edge node will query the cloud server's authentication results from the blockchain ledger to complete the re-authentication process.

It is noted that we assume that all edge nodes and vehicles authorized by the cloud server are reliable in the registration phase. IoV service providers can ensure that ISPs deploy legitimate edge nodes through legal contracts and vehicle information can be checked from the official. Besides, all interactions with the cloud go through a secure channel. While in the two-stage authentication process, the threat model mentioned in Section IV-B will happen due to the complexity of wireless transmission in mobile networks.

B. System Initialization

When the system is first startup, the cloud server $\mathcal{S}$ initializes the system parameters, which includes the prime p indicating the order of the base finite field of elliptic curve, the parameters a and b of elliptic curve, the generator G of cyclic subgroup of the elliptic curve, the prime n indicating the order of subgroup and the cofactor cf relative to n. Then, $\mathcal{S}$ generates a key-pair $(sk_{\mathcal{S}}, PK_{\mathcal{S}})$ using these parameters by ECC. The secret key $sk_{\mathcal{S}}$ is kept by $\mathcal{S}$ and never expose it to others. $\mathcal{S}$ also picks a cryptographic hash functions: $H : \{0,1\}^* \rightarrow \{0,1\}^*$. The public key $PK_{\mathcal{S}}$ along with system parameters mentioned above are open to the public. The key-pair $(sk_{\mathcal{S}}, PK_{\mathcal{S}})$ is exploited when $\mathcal{S}$ is authenticated by other entities. All subsequent key pairs are generated using these system parameters according to ECC.

C. Registration

In SEA, there are two kinds of entities that need to register to cloud server, including edge nodes and vehicles. This kind of registration is prerequisite to the subsequent authentication.

Edge node registration: Though edge nodes logically belong to the cloud server, they are physically controlled by the Internet service provider. Thus, it's necessary to prove the legitimacy of edge nodes. In registration phase, an edge node $\mathcal{E}$ generates a key-pair $(sk_{\mathcal{E}}, PK_{\mathcal{E}})$ using the system parameters. Then, it transmits the public key $PK_{\mathcal{E}}$ to $\mathcal{S}$. Upon receiving public key $PK_{\mathcal{E}}$, $\mathcal{S}$ signs on it and returns the signature $Sig_{\mathcal{S}}(PK_{\mathcal{E}})$ to $\mathcal{E}$. $\mathcal{E}$ finishes the registration as long as it receives the endorsement to its public key provided by $\mathcal{S}$.

Vehicle registration: Vehicle registration happens when a vehicle subscribes the Internet service provided by $\mathcal{S}$. In registration phase, a vehicle $\mathcal{V}$ generates a key-pair $(sk_{\mathcal{V}}, PK_{\mathcal{V}})$ using the system parameters initialized by $\mathcal{S}$, then sends its public key $PK_{\mathcal{V}}$ to $\mathcal{S}$. This key-pair $(sk_{\mathcal{V}}, PK_{\mathcal{V}})$ is exploited when $\mathcal{V}$ is authenticated by its cloud server. To confirm the legitimacy of the registered vehicle, relevant information for itself (e.g., name, driver license, car number) of the vehicle should be sent along with the public key for the IoV service provider to check.

At this moment, $\mathcal{S}$ receives the public keys $PK_{\mathcal{E}}$ and $PK_{\mathcal{V}}$ from $\mathcal{E}$ and $\mathcal{V}$, which can be exploited to authenticate $\mathcal{E}$ and $\mathcal{V}$ by verifying signatures generated by them.

D. Authentication and Key Negotiation

Before a vehicle accesses a Internet service, it should be authenticated by the service server (cloud server or edge nodes). Equally, the vehicle can also request to authenticate its serving server for security. Besides, the data transferred between the three entities may contain many sensitive information of user. Thus, three independent key agreement should be completed at the end of the authentication.

In SEA, ECDSA and ECDHE techniques are exploited respectively in authentication and key negotiation process. To sink the authentication and key negotiation capability to edge, we split a vehicle into two different authentication status, i.e., *initial authentication status* and *re-authentication status*.

Initial authentication status: Fig. 3 depicts the authentication and key agreement process while a vehicle stays in initial authentication status. The vehicle is authenticated by a cloud server, who further endorses the authentication result into channel ledger.

Authentication: When a vehicle $\mathcal{V}$ first accesses to its subscribed Internet service, the vehicle chooses a random $s_{\mathcal{V}} \in_R Z_p^*$, and computes $P_{\mathcal{V}}$ as shown in (2),

$$P_{\mathcal{V}} = s_{\mathcal{V}} \cdot G \tag{2}$$

where $s_{\mathcal{V}}$ is employed as a challenge to the edge node $\mathcal{E}$ and cloud server $\mathcal{S}$ as well as for key negotiation. In (2), the $s_{\mathcal{V}}$ and G perform the operation of scalar multiplication. Then $\mathcal{V}$ sends $P_{\mathcal{V}}$, $PK_{\mathcal{V}}$ with a service request SR to $\mathcal{E}$.

$\mathcal{E}$ chooses a random $s_{\mathcal{E}} \in_R Z_p^*$, and computes $P_{\mathcal{E}}$ as shown in (3),

$$P_{\mathcal{E}} = s_{\mathcal{E}} \cdot G \tag{3}$$

where $s_{\mathcal{E}}$ is employed as a challenge to the $\mathcal{S}$ and $\mathcal{V}$ as well as for key negotiation. $\mathcal{E}$ computes C_1 as shown in (4),

$$C_1 = H(s_{\mathcal{E}} \cdot P_{\mathcal{V}} \cdot PK_{\mathcal{V}} \cdot PK_{\mathcal{E}}) \tag{4}$$

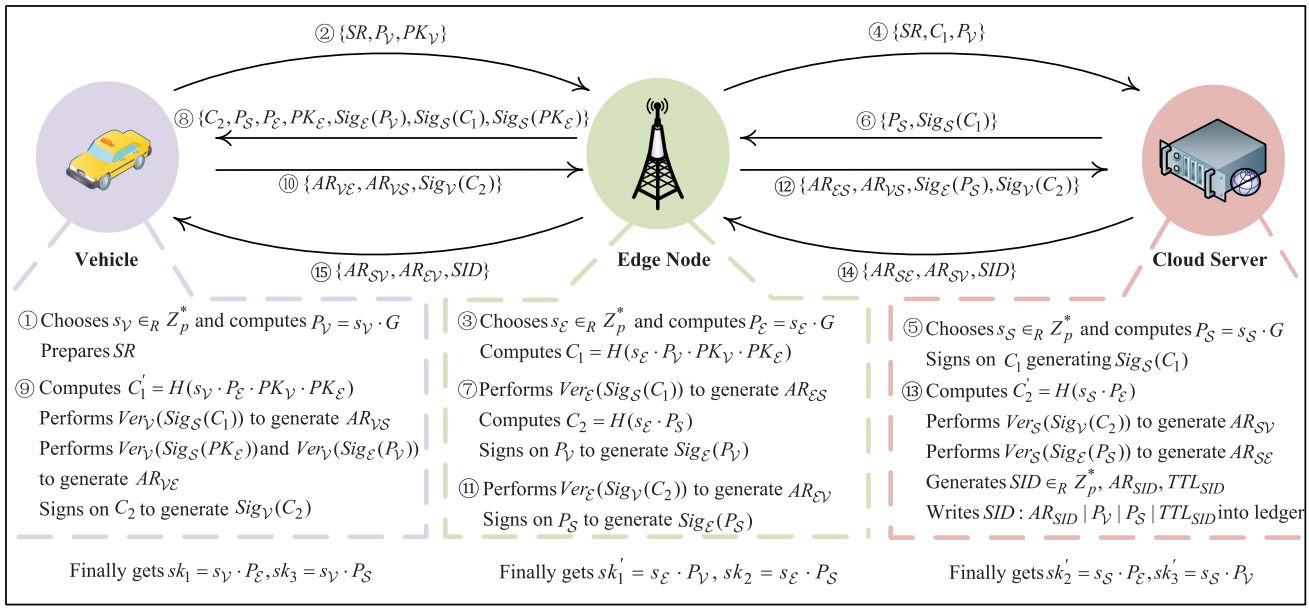

Fig. 3. Authentication and key agreement process when a vehicle first accesses a service, where it stays in initial authentication status.

where C_1 is derived from random number $s_{\mathcal{V}}$ and $s_{\mathcal{E}}$, which are temporarily generated from $\mathcal{V}$ and $\mathcal{E}$. Then, $\mathcal{V}$ sends SR, C_1 and $P_{\mathcal{V}}$ to cloud server $\mathcal{S}$.

$\mathcal{S}$ chooses a random number $s_{\mathcal{S}} \in_R Z_p^*$, and computes $P_{\mathcal{S}}$ as shown in (5),

$$P_{\mathcal{S}} = s_{\mathcal{S}} \cdot G \tag{5}$$

where $s_{\mathcal{S}}$ is employed as a challenge to $\mathcal{V}$ and $\mathcal{E}$ as well as for key negotiation. Afterwards, $\mathcal{S}$ signs on challenge message C_1 generating the signature $Sig_{\mathcal{S}}(C_1)$. Then, $\mathcal{S}$ sends $P_{\mathcal{S}}$ and $Sig_{\mathcal{S}}(C_1)$ to $\mathcal{E}$.

$\mathcal{E}$ performs $Ver_{\mathcal{E}}(Sig_{\mathcal{S}}(C_1))$ to verify the authenticity of the $\mathcal{S}$. If passes, it generates $AR_{\mathcal{ES}}$ and computes C_2 as shown in (6),

$$C_2 = H(s_{\mathcal{E}} \cdot P_{\mathcal{S}}) \tag{6}$$

where C_2 is a derived challenge message from $\mathcal{E}$ and $\mathcal{S}$ to $\mathcal{V}$. $\mathcal{E}$ signs on $P_{\mathcal{V}}$ generating signature $Sig_{\mathcal{E}}(P_{\mathcal{V}})$. Then $\mathcal{E}$ sends C_2, $P_{\mathcal{S}}, P_{\mathcal{E}}, PK_{\mathcal{E}}, Sig_{\mathcal{E}}(P_{\mathcal{V}}), Sig_{\mathcal{S}}(C_1)$ and $Sig_{\mathcal{S}}(PK_{\mathcal{E}})$ to $\mathcal{V}$, where $Sig_{\mathcal{S}}(PK_{\mathcal{E}})$ is obtained from cloud server S in registration phase.

$\mathcal{V}$ first computes C_1' as shown in (7),

$$C_1' = H(s_{\mathcal{V}} \cdot P_{\mathcal{E}} \cdot PK_{\mathcal{V}} \cdot PK_{\mathcal{E}}) \tag{7}$$

$\mathcal{V}$ performs $Ver_{\mathcal{V}}(Sig_{\mathcal{S}}(C_1))$ to verify the authenticity of $\mathcal{S}$ and generates $AR_{\mathcal{VS}}$ if it successfully verified. Then, $\mathcal{V}$ executes $Ver_{\mathcal{V}}(Sig_{\mathcal{S}}(PK_{\mathcal{E}}))$ and $Ver_{\mathcal{V}}(Sig_{\mathcal{E}}(P_{\mathcal{V}}))$ in turn to verify $\mathcal{E}$, $AR_{\mathcal{VE}}$ is generated only if both of them pass.

Finally, $\mathcal{V}$ signs on C_2 generating signature $Sig_{\mathcal{V}}(C_2)$ and sends $AR_{\mathcal{VE}}$, $AR_{\mathcal{VS}}$ and $Sig_{\mathcal{V}}(C_2)$ to $\mathcal{E}$.

$\mathcal{E}$ performs $Ver_{\mathcal{E}}(Sig_{\mathcal{V}}(C_2))$ to verify the authenticity of $\mathcal{S}$. If passes, $\mathcal{E}$ generates the authentication result $AR_{\mathcal{ES}}$. Then, $\mathcal{E}$ signs on $P_{\mathcal{S}}$ generating signature $Sig_{\mathcal{E}}(P_{\mathcal{S}})$ and sends $AR_{\mathcal{ES}}$, $AR_{\mathcal{VS}}$, $Sig_{\mathcal{E}}(P_{\mathcal{S}})$ and $Sig_{\mathcal{V}}(C_2)$ to $\mathcal{S}$.

$\mathcal{S}$ first computes C_2' as shown in (8),

$$C_2' = H(s_{\mathcal{S}} \cdot P_{\mathcal{E}}) \tag{8}$$

Then, $\mathcal{S}$ performs $Ver_{\mathcal{S}}(Sig_{\mathcal{V}}(C_2))$ and $Ver_{\mathcal{S}}(Sig_{\mathcal{E}}(P_{\mathcal{S}}))$ to respectively verify the authenticity of $\mathcal{V}$ and $\mathcal{E}$. If passes, it generates authentication results $AR_{\mathcal{SV}}$ and $AR_{\mathcal{SE}}$. Besides, $\mathcal{S}$ generates a service id $SID \in_R Z_p^*$, and writes $AR_{SID}||P_{\mathcal{V}}||P_{\mathcal{S}}||TTL_{SID}$ with SID as the key into channel ledger, where AR_{SID} is equal to $AR_{\mathcal{SV}}$. Finally, $\mathcal{S}$ sends $AR_{\mathcal{SE}}$, $AR_{\mathcal{SV}}$ and SID to $\mathcal{E}$. $\mathcal{E}$ then prepares $AR_{\mathcal{EV}}$, and forwards $AR_{\mathcal{SV}}$ and SID to $\mathcal{V}$.

Key Negotiation: Three independent session keys for any two entities will be computed. To achieve this, we make a special design on the exchanged messages during the authentication process. Specifically, the randomly generated numbers are challenges, and at the same time, are exchanged to negotiate for session keys.

At this moment, each entity has the necessary data to compute the communication secret. Specifically, $\mathcal{V}$ owns its private key $s_{\mathcal{V}}$, receives $P_{\mathcal{E}}$ from $\mathcal{E}$, and gets $P_{\mathcal{S}}$ from $\mathcal{S}$. It computes $sk_1 = s_{\mathcal{V}} \cdot P_{\mathcal{E}} = s_{\mathcal{V}} \cdot s_{\mathcal{E}} \cdot G$, and computes $sk_3 = s_{\mathcal{V}} \cdot P_{\mathcal{S}} = s_{\mathcal{V}} \cdot s_{\mathcal{S}} \cdot G$. $\mathcal{E}$ owns its private key $s_{\mathcal{E}}$, receives $P_{\mathcal{V}}$ from $\mathcal{V}$, and gets $P_{\mathcal{S}}$ from $\mathcal{S}$. It computes $sk_1' = s_{\mathcal{E}} \cdot P_{\mathcal{V}} = s_{\mathcal{E}} \cdot s_{\mathcal{V}} \cdot G = sk_1$. The negotiated secret sk_1 is exploited for encrypted communication between $\mathcal{V}$ and $\mathcal{E}$. Besides, $\mathcal{E}$ computes $sk_2 = s_{\mathcal{E}} \cdot P_{\mathcal{S}} = s_{\mathcal{E}} \cdot s_{\mathcal{S}} \cdot G$. $\mathcal{S}$ now owns its private key $s_{\mathcal{S}}$, receives $P_{\mathcal{E}}$ from $\mathcal{E}$, and gets $P_{\mathcal{V}}$ from $\mathcal{V}$. It computes $sk_2' = s_{\mathcal{S}} \cdot P_{\mathcal{E}} = s_{\mathcal{S}} \cdot s_{\mathcal{E}} \cdot G = sk_2$, and computes $sk_3' = s_{\mathcal{S}} \cdot P_{\mathcal{V}} = s_{\mathcal{S}} \cdot s_{\mathcal{V}} \cdot G = sk_3$. The negotiated secret sk_2 is exploited for encrypted communication between $\mathcal{E}$ and $\mathcal{S}$, and sk_3 is utilized for encrypted communication between $\mathcal{S}$ and $\mathcal{V}$. If the plain data M_{plain} is sensitive, $\mathcal{S}$ first encrypts M_{plain} into cipher text using sk_3 generating M_{cipher}, and then encrypts M_{cipher} again using sk_2 generating M'_{cipher}. Finally, $\mathcal{S}$ sends M'_{cipher} to $\mathcal{E}$.

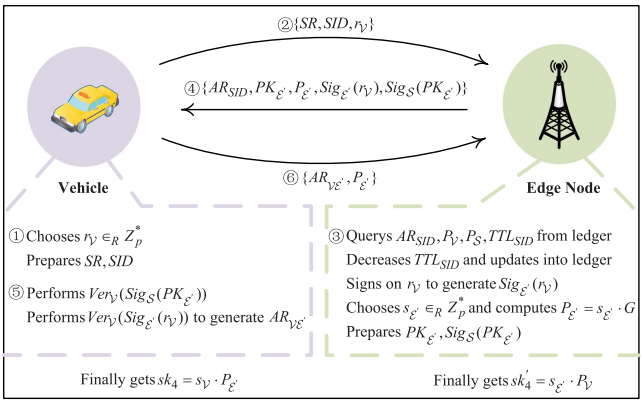

Fig. 4. Re-authentication and key agreement process when a vehicle shifts from an edge node to another, where it stays in re-authentication status.

Re-authentication status: Fig. 4 depicts the authentication and key agreement when $\mathcal{V}$ stays re-authentication status. To reduce the computation overhead and quickly respond to the vehicle, we make a dedicated design to transform the re-authentication process into a query to blockchain ledger.

Authentication: When $\mathcal{V}$ shifts from an edge node ($\mathcal{E}$) to another ($\mathcal{E}'$), it will be re-authenticated by $\mathcal{E}'$. In this case, $\mathcal{V}$ first prepares a service request SR and the service id SID (generated by $\mathcal{S}$), and chooses a random number $r_{\mathcal{V}} \in_R Z_p^*$. Then, it sends SR, SID and $r_{\mathcal{V}}$ to $\mathcal{E}'$.

Receiving SR and SID, $\mathcal{E}'$ queries the channel ledger using SID as the key. Since $\mathcal{S}$ has authenticated $\mathcal{V}$ and endorsed the result into channel ledger, the chaincode should return AR_{SID}, $P_{\mathcal{V}}$, $P_{\mathcal{S}}$ and TTL_{SID}. If the value of AR_{SID} is 1 and the value of TTL_{SID} is larger than 0, $\mathcal{E}'$ decreases TTL_{SID} and updates the new values back into channel ledger. AR_{SID} is equal to 1, which means that $\mathcal{V}$ has been authenticated by $\mathcal{S}$. TTL_{SID} is larger than 0, which means that AR_{SID} is still valid. Otherwise, $\mathcal{E}'$ terminates the communication. In such condition, $\mathcal{V}$ has to switch into the initial authentication phase. In other words, $\mathcal{V}$ should be authenticated by $\mathcal{S}$. Then, $\mathcal{E}'$ signs on the received $r_{\mathcal{V}}$ generating $Sig_{\mathcal{E}'}(r_{\mathcal{V}})$. Besides, $\mathcal{E}'$ chooses $s_{\mathcal{E}'} \in_R Z_p^*$, and computes $P_{\mathcal{E}'}$ as shown in (9),

$$P_{\mathcal{E}'} = s_{\mathcal{E}'} \cdot G \tag{9}$$

Finally, $\mathcal{E}'$ sends AR_{SID}, $P_{\mathcal{E}'}$, $Sig_{\mathcal{E}'}(r_{\mathcal{V}})$ with its local $PK_{\mathcal{E}'}$ and $Sig_{\mathcal{S}}(PK_{\mathcal{E}'})$ to $\mathcal{V}$.

$\mathcal{V}$ first performs $Ver_{\mathcal{V}}(Sig_{\mathcal{S}}(PK_{\mathcal{E}'}))$ to verify the authenticity of the public key $PK'_{\mathcal{E}'}$ of $\mathcal{E}'$. If passes, it further performs $Ver_{\mathcal{V}}(Sig_{\mathcal{E}'}(r_{\mathcal{V}}))$ to verify the authenticity of $\mathcal{E}'$. If passes, $\mathcal{V}$ generates authentication results $AR_{\mathcal{V}\mathcal{E}'}$, and then sends $AR_{\mathcal{V}\mathcal{E}'}$ and $P_{\mathcal{E}'}$ to $\mathcal{E}'$. $\mathcal{E}'$ will check the $P_{\mathcal{E}'}$ received. If $\mathcal{E}'$ needs to further communicate with $\mathcal{S}$, $\mathcal{E}'$ and $\mathcal{S}$ perform a mutually challenge-response action similar to the pattern in initial authentication status. Extra challenges are not necessary to generate, since $P_{\mathcal{V}}$ and $P_{\mathcal{S}}$ can be reused.

Key Negotiation: At this moment, both $\mathcal{V}$ and $\mathcal{E}'$ have mutually authenticate themselves to each other and have necessary data to computes the communication secret. Specifically, $\mathcal{V}$ owns its private key $s_{\mathcal{V}}$, receives $P_{\mathcal{E}'}$ from $\mathcal{E}'$. It computes $sk_4 = s_{\mathcal{V}} \cdot P_{\mathcal{E}'} = s_{\mathcal{V}} \cdot s_{\mathcal{E}'} \cdot G$. $\mathcal{E}'$ owns its private key $s_{\mathcal{E}'}$, and gets $P_{\mathcal{V}}$ and $P_{\mathcal{S}}$ from the blockchain ledger. It computes $sk'_4 = s_{\mathcal{E}'} \cdot P_{\mathcal{V}} = s_{\mathcal{E}'} \cdot s_{\mathcal{V}} \cdot G = sk_4$, and computes $sk_5 = s_{\mathcal{E}} \cdot P_{\mathcal{S}} = s_{\mathcal{E}} \cdot s_{\mathcal{S}} \cdot G$. The secret sk_4 is exploited for encrypted communication between $\mathcal{V}$ and $\mathcal{E}'$. $\mathcal{S}$ owns its private key $s_{\mathcal{S}}$, and receives $P_{\mathcal{E}'}$ from $\mathcal{E}'$, it computes $sk'_5 = s_{\mathcal{S}} \cdot P_{\mathcal{E}'} = s_{\mathcal{S}} \cdot s_{\mathcal{E}'} \cdot G$. Secret sk_5 is used for encrypted communication between $\mathcal{S}$ and $\mathcal{E}'$. It is noted that the secret sk_3 can still be exploited for encrypted communication between $\mathcal{S}$ and $\mathcal{V}$. In re-authentication status, $\mathcal{V}$ is authenticated by $\mathcal{E}'$ without the participation of $\mathcal{S}$, which eliminates the network communication latency.

VI. Security Analysis

In this section, we give the security analysis to prove whether each security objective mentioned in Section IV-C2 has been satisfied to demonstrate the security of SEA.

There are three types of entities in SEA. $\mathcal{S}$ keeps secret keys $s_{\mathcal{S}}$ and $sk_{\mathcal{S}}$. $\mathcal{E}$ keeps secret keys $s_{\mathcal{E}}$ and $sk_{\mathcal{E}}$. $\mathcal{V}$ keeps secret keys $s_{\mathcal{V}}$ and $sk_{\mathcal{V}}$. Any of the involved entities can be treated as an oracle that usually has the following two status: *Accept*, which means the oracle receives a right message; *Reject*, which means the oracle receives a fault message.

Definition 1 (Adversary's Capability): An adversary $\mathcal{A}$ can perform operations as follows to attack authentication system:

- Exchanged-Message-Capture: This operation can be exploited to capture the message transmitted over the network such as signatures and public keys during authentication process. Attackers may try getting useful information from these messages, so as to carry out impersonate attacks. He/She can also replay this message directly for illegal authorization.
- Exchanged-Key-Query: This query is exploited to reveal the negotiated private key of entities. The adversay $\mathcal{A}$ can invoke this query to entities with an exchanged key $s_{\mathcal{A}} \cdot P$. The target entity replies with an another exchanged key $s_{Entity} \cdot P$, where s_{Entity} is secretly kept by the entity.

Theorem 1: SEA scheme can guarantee that $\mathcal{S}$, $\mathcal{E}$ and $\mathcal{V}$ are authenticated by any other entities, and adversaries $\mathcal{A}^* = (\mathcal{A}_{\mathcal{S}}, \mathcal{A}_{\mathcal{E}}, \mathcal{A}_{\mathcal{V}})$ will be rejected.

Proof: We assume that $\mathcal{A}_{\mathcal{V}}$, $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{S}}$ are adversaries who try to impersonate entities by cheating SEA with unauthorized public key and signature forged based on it. $\mathcal{A}_{\mathcal{V}}$ and $\mathcal{A}_{\mathcal{E}}$ have no reliable public key recorded in $\mathcal{S}$.

In initial authentication status, the message C_1 sent to $\mathcal{S}$ is derived from challenge messages $s_{\mathcal{V}}$ and $s_{\mathcal{E}}$ randomly generated by $\mathcal{V}$ and $\mathcal{E}$ respectively, since $C_1 = H(s_{\mathcal{E}} \cdot P_{\mathcal{V}} \cdot PK_{\mathcal{V}} \cdot PK_{\mathcal{E}}) = H(s_{\mathcal{E}} \cdot s_{\mathcal{V}} \cdot P \cdot PK_{\mathcal{V}} \cdot PK_{\mathcal{E}})$. It's known that $PK_{\mathcal{S}}$ is open to public, $\mathcal{S}$ is supposed to sign on C_1 so that $\mathcal{V}$ and $\mathcal{E}$ can independently authenticate $\mathcal{S}$ by performing $Ver_{\mathcal{V}}(Sig_{\mathcal{S}}(C_1))$ and $Ver_{\mathcal{E}}(Sig_{\mathcal{S}}(C_1))$. $\mathcal{A}_{\mathcal{S}}$ who wants to impersonate $\mathcal{S}$ can't calculate message C_1 since $s_{\mathcal{V}}$ and $s_{\mathcal{E}}$ are kept by their owning entities. Even if $\mathcal{A}_{\mathcal{S}}$ can intercept transmitted message C_1, he/she is still unable to generate signature on C_1 by private $sk_{\mathcal{S}}$. $\mathcal{A}_{\mathcal{S}}$ will be rejected by SEA system at last. The message C_2 sent to $\mathcal{V}$ is derived from challenge $s_{\mathcal{E}}$ and $s_{\mathcal{S}}$ randomly generated by $\mathcal{E}$ and $\mathcal{S}$ respectively, since $C_2 = H(s_{\mathcal{E}} \cdot P_{\mathcal{S}}) = H(s_{\mathcal{E}} \cdot s_{\mathcal{S}} \cdot P)$. $\mathcal{E}$ and $\mathcal{S}$

TABLE III
COMPARISON OF TIME-CONSUMING CRYPTOGRAPHIC OPERATIONS WITH DIFFERENT SCHEMES

Scheme		Vehicle	Edge Node	Cloud Server
SEA	Initial Authentication	$14SM$	$11SM$	$8SM$
	Re-authentication	$5SM$	$2SM$	–
RCoM [12]		$2Pair + 9Exp + 5PM$	$2Pair + 5Exp + 4PM$	$4Pair + 3Exp + 3PM$
ECBPA[24]		$8SM$	$9SM$	$3SM$

can dependently authenticate $\mathcal{V}$ by performing $Ver_{\mathcal{E}}(Sig_{\mathcal{V}}(C_2))$ and $Ver_{\mathcal{S}}(Sig_{\mathcal{V}}(C_2))$. Since the public key of vehicle has been registered, $\mathcal{A}_{\mathcal{V}}$ cannot utilize his/her own key pair to fabricated signature to deceive $\mathcal{S}$. Besides, he/she also has no means of getting $sk_{\mathcal{V}}$. $\mathcal{V}$ authenticates $\mathcal{E}$ by perform $Ver_{\mathcal{V}}(Sig_{\mathcal{S}}(PK_{\mathcal{E}}))$ and $Ver_{\mathcal{V}}(Sig_{\mathcal{E}}(P_{\mathcal{V}}))$, where message $P_{\mathcal{V}} = s_{\mathcal{V}} \cdot P$, $s_{\mathcal{V}}$ is randomly generated by $\mathcal{V}$. It's noted that $Sig_{\mathcal{S}}(PK_{\mathcal{E}})$ is the signature of $\mathcal{S}$ on legitimate $\mathcal{E}$'s public key during registration phase. Without it, adversary $\mathcal{A}_{\mathcal{E}}$ may pass the verification by sending $PK_{\mathcal{A}_{\mathcal{E}}}$ and $Sig_{\mathcal{A}_{\mathcal{E}}}(P_{\mathcal{V}})$ to $\mathcal{V}$ since $PK_{\mathcal{E}}$ is sent along with $Sig_{\mathcal{E}}(P_{\mathcal{V}})$. By executing $Ver_{\mathcal{V}}(Sig_{\mathcal{S}}(PK_{\mathcal{E}}))$, $\mathcal{V}$ rules out the invalid public key and refuse $\mathcal{A}_{\mathcal{E}}$. $\mathcal{S}$ authenticates $\mathcal{E}$ by perform $Ver_{\mathcal{S}}(Sig_{\mathcal{E}}(P_{\mathcal{S}}))$ with registered $PK_{\mathcal{E}}$, where message $P_{\mathcal{S}} = s_{\mathcal{S}} \cdot P$, $s_{\mathcal{S}}$ is randomly generated by $\mathcal{S}$.

In re-authentication status, it need to achieve authentication between $\mathcal{V}$ and $\mathcal{E}'$. $\mathcal{E}'$ can authenticate $\mathcal{V}$ by verifying the validity of AR_{SID} recorded on blockchain ledger which $\mathcal{A}_{\mathcal{V}}$ is unable to tamper with. $\mathcal{V}$ authenticate $\mathcal{E}'$ by performing $Ver_{\mathcal{V}}(Sig_{\mathcal{S}}(PK_{\mathcal{E}'}))$ and $Ver_{\mathcal{V}}(Sig_{\mathcal{E}'}(r_{\mathcal{V}}))$. $Sig_{\mathcal{S}}(PK_{\mathcal{E}'})$ is sent by $\mathcal{S}$ during the registration phase, which indicates the cloud server's recognition of public key $PK_{\mathcal{E}'}$. Since $\mathcal{A}_{\mathcal{E}}$ has no means of getting $Sig_{\mathcal{S}}(PK_{\mathcal{A}_{\mathcal{E}}})$, when it sends $PK_{\mathcal{A}_{\mathcal{E}}}$ and $Sig_{\mathcal{A}_{\mathcal{E}}}(r_{\mathcal{V}})$ to $\mathcal{V}$, $PK_{\mathcal{A}_{\mathcal{E}}}$ will be judged invalid.

Thus, SEA guarantees that $\mathcal{S}$, $\mathcal{E}$ and $\mathcal{V}$ are authenticated by any other entities. $\mathcal{A}_{\mathcal{V}}$, $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{S}}$ are expected to be refused to join SEA system.

Theorem 2: SEA scheme can negotiate three independent session keys, which are exploited by different two entities, and adversary $\mathcal{A}$ cannot infer any of these session keys.

Proof: In initial authentication status, $\mathcal{V}$ receives $P_{\mathcal{E}}$ from $\mathcal{E}$, gets $P_{\mathcal{S}}$ from $\mathcal{S}$, computes $sk_1 = s_{\mathcal{V}} \cdot P_{\mathcal{E}} = s_{\mathcal{V}} \cdot s_{\mathcal{E}} \cdot G$, and computes $sk_3 = s_{\mathcal{V}} \cdot P_{\mathcal{S}} = s_{\mathcal{V}} \cdot s_{\mathcal{S}} \cdot G$. $\mathcal{E}$ receives $P_{\mathcal{V}}$ from $\mathcal{V}$, gets $P_{\mathcal{S}}$ from $\mathcal{S}$, computes $sk'_1 = s_{\mathcal{E}} \cdot P_{\mathcal{V}} = s_{\mathcal{E}} \cdot s_{\mathcal{V}} \cdot G$, and computes $sk_2 = s_{\mathcal{E}} \cdot P_{\mathcal{S}} = s_{\mathcal{E}} \cdot s_{\mathcal{S}} \cdot G$. sk_1 is equal to sk'_1, which is exploited by $\mathcal{V}$ and $\mathcal{E}$. $\mathcal{S}$ receives $P_{\mathcal{V}}$ from $\mathcal{V}$, gets $P_{\mathcal{E}}$ from $\mathcal{E}$, computes $sk'_2 = s_{\mathcal{S}} \cdot P_{\mathcal{E}} = s_{\mathcal{S}} \cdot s_{\mathcal{E}} \cdot G$, and computes $sk'_3 = s_{\mathcal{S}} \cdot P_{\mathcal{V}} = s_{\mathcal{S}} \cdot s_{\mathcal{V}} \cdot G$. sk_3 is equal to sk'_3, which is exploited between $\mathcal{V}$ and $\mathcal{S}$.

In re-authentication status, $\mathcal{V}$ computes $sk_4 = s_{\mathcal{V}} \cdot P_{\mathcal{E}'} = s_{\mathcal{V}} \cdot s_{\mathcal{E}'} \cdot G$. $\mathcal{E}$ computes $sk'_4 = s_{\mathcal{E}'} \cdot P_{\mathcal{V}} = s_{\mathcal{E}'} \cdot s_{\mathcal{V}} \cdot G$, and computes $sk_5 = s_{\mathcal{E}'} \cdot P_{\mathcal{S}} = s_{\mathcal{E}'} \cdot s_{\mathcal{S}} \cdot G$. $\mathcal{S}$ computes $sk'_5 = s_{\mathcal{S}} \cdot P_{\mathcal{E}'} = s_{\mathcal{S}} \cdot s_{\mathcal{E}'} \cdot G$. There is no need for $\mathcal{V}$ and $\mathcal{S}$ to negotiate a new session key, since sk_3 is still available.

The secret keys $s_{\mathcal{V}}$, $s_{\mathcal{E}}/s_{\mathcal{E}'}$ and $s_{\mathcal{S}}$ are kept only by $\mathcal{V}$, $\mathcal{E}/\mathcal{E}'$ and $\mathcal{S}$. $\mathcal{A}$ can get any messages that exchanged among the entities when they are negotiating session keys. However, $\mathcal{A}$ cannot compute deduce session keys as long as the ECHD problem are unsolvable.

Theorem 3: SEA scheme can preserve $\mathcal{V}$ the privacy of sensitive data (e.g., personal settings, browsing history) from curious $\mathcal{E}$.

Proof: As mentioned above, $\mathcal{V}$ and $\mathcal{S}$ communicate with each other using an independent session key. Assumed M_{plain} is some sensitive data, $\mathcal{V}$ first encrypts it into cipher text M_{cipher} using sk_3. Then $\mathcal{V}$ further encrypts M_{cipher} into M'_{cipher} using sk_1/sk_4, and sends M'_{cipher} $\mathcal{E}/\mathcal{E}'$. $\mathcal{E}/\mathcal{E}'$ can easily decrypt M'_{cipher} getting M_{cipher}. However, it cannot retrieve M_{plain}, since $\mathcal{E}/\mathcal{E}'$ has no knowledge of sk_3.

Theorem 4: SEA scheme can provide perfect forward secrecy to the involved entities.

Proof: Assume that sk_{old} is a session key used by $\mathcal{V}$ before, and $sk_{current}$ is a current session key for $\mathcal{V}$. $sk_{current}$ is leaked to $\mathcal{A}$, so he can decrypt the ciphertext encrypted by $sk_{current}$. Since $sk_{old} = s_{\mathcal{V}} \cdot P_{\mathcal{E}}$, sk_{old} depends on $s_{\mathcal{V}}$. It is infeasible for $\mathcal{A}$ to retrieve messages encrypted by sk_{old}, since he cannot deduce the private key $s_{\mathcal{V}}$ as long as the ECDH is unsolvable. Every time $\mathcal{V}$ accesses an IoV service or shifts to a new edge node, it negotiates new session keys with $\mathcal{S}$ and $\mathcal{E}/\mathcal{E}'$. The session keys used by $\mathcal{E}/\mathcal{E}'$ and $\mathcal{S}$ also hold. Thus, SEA provides perfect forward secrecy to the involved entities.

Theorem 5: SEA scheme can defend against replay attacks, adversaries $\mathcal{A}^* = (\mathcal{A}_{\mathcal{S}}, \mathcal{A}_{\mathcal{E}}, \mathcal{A}_{\mathcal{V}})$ can not be accepted by replaying the message sent before.

Proof: The authentication process in our SEA employing the mode of challenge-response mode, which is an effective way to defend against replay attacks. Assuming that $\mathcal{A}_{\mathcal{V}}$ captures the message sent to the edge nodes when a vehicle executing the protocol before. $\mathcal{A}_{\mathcal{V}}$ firstly sends the message $\{SR, P_{\mathcal{V}}, PK_{\mathcal{V}}\}$ to edge node. Upon receiving it, the edge node randomly chooses a number $s_{\mathcal{E}} \in_R Z_p^*$ and compute the $C_2 = H(s_{\mathcal{E}} \cdot P_{\mathcal{S}})$, which will be send to the requested vehicle. $s_{\mathcal{E}}$ is remained as a challenge message. In normal case, the vehicle will sign on C_2 using $sk_{\mathcal{V}}$ and send $Sig_{\mathcal{V}}(C_2)$ back to edge node. The edge node can authenticate the vehicle by performing $Ver_{\mathcal{E}}(Sig_{\mathcal{V}}(C_2))$. But in the replay attack scenario, since $s_{\mathcal{E}}$ is regenerated in every authentication, C_2 is naturally different from before. $\mathcal{A}_{\mathcal{V}}$ cannot replay the previous signature to edge node to pass the verification. And the attacker does not have $sk_{\mathcal{V}}$, so he also cannot generate a signature on latest challenge message C_2. The adversary finally is refused by SEA authentication system.

Similarly, the C_1 computed including the changeable challenge messages $s_{\mathcal{V}}$ and $s_{\mathcal{E}}$. By verifying $Sig_{\mathcal{S}}(C_1)$, $\mathcal{V}$ and $\mathcal{E}$ can authenticate $\mathcal{S}$. $\mathcal{A}_S$ who replays the previous signature on C_1 will be denied. And by generating the signature on time-varying $P_{\mathcal{V}}$ and $P_{\mathcal{S}}$, the $\mathcal{E}$ can be authenticated by $\mathcal{V}$ and $\mathcal{S}$. In re-authentication process, random numbers $r_{\mathcal{V}}$ and $s_{\mathcal{E}'}$ are used

as challenge information to prevent replay attacks. The adversary cannot pass the authentication by replaying $Sin_{\mathcal{E}'}(r_{\mathcal{V}})$ or $P_{\mathcal{E}'}$.

VII. Performance Evaluation

In this section, we conduct extensive experiments to evaluate the performance of SEA and compare it with the state-of-the-art methods.

A. Experimental Settings

1) Prototype Implementation: We simulate vehicle operations on a notebook with Intel(R) Core(TM) i5-8250 CPU @1.60 GHz and 8 GB physical memory. Operations related to the edge node are executed on a notebook with Intel(R) Core(TM) i5-4200 CPU @1.70 GHz and 8 GB physical memory. The cloud server runs on a PC with AMD Ryzen 5 2600X CPU @3.60 GHz and 16 GB memory. For each device, we install the Windows 10 Pro 64-bit operating system and prepare the Java version 1.8.0_261 (Java(TM) SE Runtime Environment, build 1.8.0_261-b12) for running java codes.

To evaluate the latency introduced by consortium blockchain, we build a consortium blockchain with several peer nodes in a virtual machine of Ubuntu 16.04.05 hosted on VMware Workstation 15 Pro with Fabric 1.3. We implement SEA using the elliptic curve $secp256r1$ and Java $JPBC$ 2.0.0 library. The secret size is set to 256 bits.

2) Methods to Compare: We compare SEA with two types of schemes, which are cloud-based authentication scheme RCoM [29] and proxy-based authentication scheme ECBPA [30]. Both of them involve secure and efficient authentication of vehicles and have the same IoV architecture composition as our solution, so we choose them as the comparison schemes. In RCoM, the vehicle and the roadside unit (RU) are authorized by the root authority (RA). RU manages entered vehicles through token distribution. In ECBPA, the vehicle is pre-assigned secret and long-term certificate by TA. And TA then select and authenticate edge computing vehicles which will verify ordinary vehicles by group batch authentication. We do not compare SEA with the blockchain-based schemes, since blockchain is mostly a supplement to achieving trust management and will not bring many efficiency gains.

It is noted that both RCoM and ECBPA perform the same authentication process every time the service handover happens, while SEA performs the initial authentication process when the vehicle first accesses service and afterwards performs the re-authentication process.

B. Evaluation on Time Efficiency of Authentication

In this subsection, we make theoretical analysis and experimental evaluation to compare the time efficiency of SEA with the related schemes, i.e., RCoM and ECBPA.

1) Theoretical Analysis: To theoretically analyze the efficiency of the authentication scheme, we abstract time-consuming cryptographic operations, which are dominant in determining the time overhead of the scheme. More specifically, as depicted in Table III, we select 4 types of operations, namely point multiplication (PM), scalar multiplication (SM), exponentiation (Exp) and bilinear pairing ($Pair$). Then, we count the occurrence of these operations in two authentication stage of SAE.[1] And for RCoM and ECBPA, authentication and re-authentication are exactly the same process, so their counting results involve only one authentication process. Prior knowledge from practical tests shows that $Pair$ operation takes the most time, followed by Exp and SM.

We can find that SEA is more efficient than RCoM, but costs more than ECBPA when a vehicle stays in initial authentication status. When it turns into re-authentication status, SEA takes much less time than both RCoM and ECBPA for the re-authentication process. Considering that re-authentication happens more frequently compared to initial authentication which happens only once, the proposed scheme outperforms the other two schemes in terms of the overall time efficiency.

2) Experimental Analysis: To verify the correctness of the theoretical analysis, we further make experiments to compare the total time for authentication with the three different schemes. We conduct 50 tests and record the average time for each scheme in Table IV.

For each scheme, the authentication time mainly consists of 4 parts, including the time spent on operations at vehicle, edge node, and cloud server, respectively, and the communication latency between edge node and cloud server.[2] We record two stage authentication time of SEA and one authentication time of RCoM and ECBPA. Note that we neglect the communication latency between the vehicle and the edge node, as it is around 1 ms in 5 G network and far more less than the latency traversing the core network (around 184.43 ms).

From the Total Time column in Table IV, we get the observation that although SEA is slightly inferior to both RCoM and ECBPA in the initial authentication process, its re-authentication process outperforms these two schemes. We analyze the reasons for the results. First, when performing the initial authentication, the edge node and the cloud server involved in SEA have to make interaction twice, which can be observed clearly from Fig. 3. Thus, the total time of SEA includes more network communication latency than the other two schemes. Second, in the re-authentication process, we find that the total time of SEA is about one-half of that of ECBPA and one-quarter of that of RCoM. In fact, it is the non-participation of the cloud server that significantly reduces the time of re-authentication in SEA. Just by reading the authentication results from the blockchain ledger, the new edge node can achieve re-authentication of the vehicle. Due to that the re-authentication occurs more often in real scenarios of IoV, SEA is superior to RCoM and ECBPA.

Moreover, we find that the experimental results of the three entities' time are consistent with the theoretical analysis, which proves its correctness. We focus on the re-authentication of SEA to explain the correspondence between experiment and theory. The vehicle takes less time than RCoM and ECBPA, since it only needs to authenticate the new edge node and undertakes fewer

[1] It is noted that some operations occur more than once in a single entity (e.g., $s_{\mathcal{V}} \cdot P_{\mathcal{E}}$ in the vehicle) in SEA, which can be reused in execution and are counted only once.

[2] To obtain the average round-trip time between an edge node and the cloud server, we select the Top-10 K websites in the Alexa ranking as cloud servers and get the average latency as 184.43 ms.

TABLE IV
COMPARISON OF AUTHENTICATION TIME WITH DIFFERENT SCHEMES (UNITS: MS)

Scheme		Vehicle	Edge Node		Cloud Server	E2C Latency	Total Time
			Ledger Query	Computation			
SEA	Initial Authentication	12.15	–	9.56	6.46	368.86	397.03
	Re-Authentication	4.14	76.30	2.24	–	–	82.68
RCoM		67.33	–	39.99	27.93	184.43	319.68
ECBPA		5.54	–	5.83	1.92	184.43	197.72

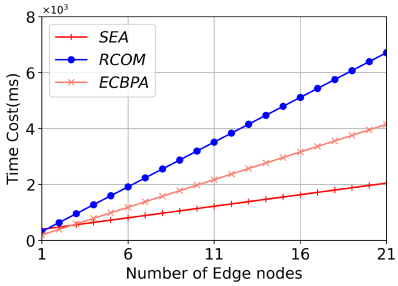

Fig. 5. Comparison of the total time spent on authentication with a varying number of edge nodes on a vehicle's route.

TABLE V
COMPARISON OF COMMUNICATION OVERHEAD OF DIFFERENT SCHEMES (UNIT: BYTE)

Scheme		Vehicle	Edge Node	Cloud Server
SEA	Initial Authentication	195	741	162
	Re-authentication	130	257	–
RCoM [12]		876	268	602
ECBPA [24]		524	424	160

operations. Regardless of the time of reading the ledger, the edge node that undertakes fewer SM operations spends the least time. And the cloud server spends no time due to its non-participation in the re-authentication.

To further illustrate the superiority of our SEA, we also evaluate the overall time of the three schemes in certain scenarios. In edge-integrated IoV, the vehicle moves fast and leads to frequent service handovers from one edge node to another, which leads to repetitive authentication processes. We track the authentication time of this process. As shown in Fig. 5, compared with other schemes, SEA spends less time on authentication and its superiority becomes more significant as the number of edge nodes along a vehicle's route increases. This is because RCoM and ECBPA have to suffer the network communication latency every time the vehicle switches an edge node area, while SEA only needs to query the blockchain to finish the re-authentication process without interacting with the cloud server. It indicates that SEA is capable of shortening the delay for authentication and thereby ensures the continuity of IoV services.

C. Evaluation on Communication Overhead

In this subsection, we compare the communication overhead of different schemes, as summarized in Table V.

For an initial authentication with SEA, the cloud server sends out $P_{\mathcal{S}}$, $Sig_{\mathcal{S}}(C_1)$, $AR_{\mathcal{SE}}$, $AR_{\mathcal{SV}}$ and SID, which are totally 162 bytes. The edge node sends out 741 bytes, including SR, C_1, $P_{\mathcal{V}}$, C_2, $P_{\mathcal{S}}$, $P_{\mathcal{E}}$, $PK_{\mathcal{E}}$, $Sig_{\mathcal{E}}(P_{\mathcal{V}})$, $Sig_{\mathcal{S}}(C_1)$, $AR_{\mathcal{SV}}$, $AR_{\mathcal{ES}}$, $AR_{\mathcal{VS}}$, $Sig_{\mathcal{E}}(P_{\mathcal{S}})$, $Sig_{\mathcal{S}}(PK_{\mathcal{E}})$, $Sig_{\mathcal{V}}(C_2)$, $AR_{\mathcal{EV}}$ and SID. The Vehicle sends out SR, $P_{\mathcal{V}}$, $PK_{\mathcal{V}}$, $AR_{\mathcal{VE}}$, $AR_{\mathcal{VS}}$ and $Sig_{\mathcal{V}}(C_2)$, which are 195 bytes in total. For a re-authentication, the edge node sends out AR_{SID}, $PK_{\mathcal{E}'}$, $P_{\mathcal{E}'}$, $Sig_{\mathcal{E}'}(r_{\mathcal{V}})$ and $Sig_{\mathcal{S}}(PK_{\mathcal{E}'})$, which are 257 bytes in total. The vehicle sends out SR, SID, $r_{\mathcal{V}}$, $AR_{\mathcal{E}'}$ and $P_{\mathcal{E}'}$ which are 130 bytes. The cloud server is not involved in re-authentication, which is considered to have no communication overhead.

In RCoM [29], the vehicle, edge node (RU_ℓ in [29]) and cloud server needs to transmit 876 bytes, 268 bytes and 602 bytes, respectively.[3] In ECBPA [30], the vehicle, edge node (ECV_i in [30]) and cloud server (TA in [30]) consumes bandwidth of 524 bytes, 424 bytes and 160 bytes, respectively.[4]

Table V shows that for the initial authentication process, SEA has the least communication overhead. The advantage of SEA over the other two schemes is more significant for the re-authentication process, as SEA can greatly reduce the total amount of massage sent among these entities. In particular, SEA can reduce the heavy burden on vehicle side during re-authentication, e.g., from 876 Bytes with RCoM or 524 Bytes with ECBPA to 130 Bytes with SEA.

TABLE VI
BLOCKCHAIN-INTRODUCED LATENCY IN SEA (UNIT: MS)

Blockchain Operation	Number of Peer Nodes			
	4	8	12	16
Chaincode Read	77.63	75.83	76.42	77.06
Chaincode Write	282.86	340.03	393.43	430.57

D. Evaluation on Scalability

SEA employs consortium blockchain as a fundamental component to facilitate re-authentication process, where an edge node authenticates a vehicle by querying the authentication result from channel ledger. Afterwards, the edge node updates the values of the returned record and writes it back to channel

[3]Vehicle V_j sends tuple $\mathcal{T}_j = (SA_i, V_j, vsk_{j,1}, vsk_{j,2}, t_j, \theta_{j,1}, \theta_{j,2})$ to RU_ℓ, and sends $U = (u_1, u_2, u_3, u_4)$ and $W = (SA_i, SA_\ell, V_j, vsk_{j,1}, vsk_{j,2}, t_l, t_j, \mathbb{T}_d, \theta_l, \text{Time})$ to cloud server. RU_ℓ sends authentication tuple $\mathcal{T}_l = (SA_\ell, rsk_{l,1}, rsk_{l,2}, t_l, \mathbb{T}_d, \theta_l)$ to V_j. The cloud server sends tuple (U, W) to RA. Identities of entities, timestamp are assumed to be 4-byte, and uncertain variables (e.g., road condition information I are not considered.)

[4]TA sends D, Y, Z_{TA}, T_{TA} to ECV_i. EVC_i sends K_{ECV}, T_{ECV} to TA, and sends PID_i, W_i, F_i, G_i, U_i, T_i to V_j. V_j sends $2PID_j$, W_j, F_j, $2T_j$, E_j, tt_j, M_j, Y_j, S_j to ECV_i. Timestamp is assumed to be 4-byte, and uncertain variables (e.g., M_j) are not counted.

ledger. SEA has to suffer one chaincode read and one chaincode write in re-authentication process.

We conduct the experiment on the basis of *first-network* in the *fabric-samples*.

The *BatchSize* parameter is set to 0.05 s. We gradually scale the network by increasing its peer nodes from 4 to 16, whose results are recorded in Table VI. It indicates that the chaincode read cost is about 76 ms and fluctuates in a small amplitude. Essentially, the chaincode read operation is a process of reading data field from a local file, since every peer node has a local copy of the channel ledger.

Although the results show that chaincode write takes more time to achieve the global consensus status as the peer nodes increase, it has no influence on the time spent on re-authentication as depicted in Table IV. We elaborate on the reason as follows. An edge node usually covers around 300-1000m [52]. Assume that a vehicle runs at an extremely high speed of 160 km/h (which surpasses most countries' expressway speed limits [53], we can roughly evaluate the duration a vehicle shits from an edge node to the next one, by computing $\frac{300\,\mathrm{m}}{160{,}000\,\mathrm{m}/36000\,\mathrm{s}} = 6.75\,\mathrm{s}$. It means that when a vehicle finishes an area handover, the previous updated record to authentication result has already achieved global consensus status. Consequently, in re-authentication process, only one chaincode read cost that equals to a read to a local file should be put into consideration, which greatly improves time efficiency as the number of service handovers increases. Therefore, SEA can maintain its performance with a varying number of peer nodes in blockchain.

VIII. Discussion

By exploiting blockchain, SEA sinks the authentication process into edge, which eliminates the network communication latency. We discuss it further in this section.

SEA has good scalability in supporting diverse Internet services. In SEA, data belonging to different Internet services are isolated from dependent channels. And different channel ledgers [54] are shared by a group of nodes/clients with permission.

SEA can be influenced by some conventional attacks such as resource drain attack and pre-computation attack. We give some advice which may mitigate them. For the resource drain attack, the server deployer can reduce the loss by configuring high-defense servers and server cluster. Recently, artificial intelligence [55] can also be applied to defend it. For the pre-computation attack, the process of pre-computation is time-consuming and the cost of it grows with the increase of security level. In the future, We can entrust a reliable platform to generate key pairs so that elliptic curve parameters are not exposed to the adversary. Moreover, the security of V2V (vehicle-to-vehicle) communication is not considered in this paper. In the future, we can redesign SEA to make it compatible with V2V mutual authentication.

IX. Conclusion

In this paper, we proposed a secure and efficient blockchain-assisted authentication scheme for edge-integrated IoV. SEA achieves mutual authentication among vehicles, edge nodes and cloud servers. The cloud server only involves in initial authentication process and records authentication result in the blockchain. Edge nodes re-authenticate vehicles by querying channel ledgers, which significantly reduces computation overhead and eliminates network communication delay. Besides, a dedicated key agreement mechanism is designed to negotiate three independent session keys for any two of involved entities, which can also be utilized to protect sensitive data privacy of vehicles from being peeked by curious edge nodes. Extensive experiments have been conducted to show the effectiveness and efficiency of SEA. In the future, we will focus on integrating the authentication scheme with fundamental security functionalities provided by Internet service providers.

References

[1] W. Duan, J. Gu, M. Wen, G. Zhang, Y. Ji, and S. Mumtaz, "Emerging technologies for 5G-IoV networks: Applications, trends and opportunities," *IEEE Netw.*, vol. 34, no. 5, pp. 283–289, Sep./Oct. 2020.

[2] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, Oct. 2018.

[3] W. H. Hassan et al., "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, 2019.

[4] M. R. Dey, S. Sharma, R. C. Shit, C. P. Meher, and H. K. Pati, "IoV based real-time smart traffic monitoring system for smart cities using augmented reality," in *Proc. Int. Conf. Vis. Towards Emerg. Trends Commun. Netw.*, 2019, pp. 1–6.

[5] S. Sattar, H. K. Qureshi, M. Saleem, S. Mumtaz, and J. Rodriguez, "Reliability and energy-efficiency analysis of safety message broadcast in VANETs," *Comput. Commun.*, vol. 119, pp. 118–126, 2018.

[6] S. Kumar, S. Gollakota, and D. Katabi, "A cloud-assisted design for autonomous driving," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 41–46.

[7] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, "Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: Overview, design, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 62–68, Jan. 2017.

[8] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7702–7712, Oct. 2019.

[9] G. De La Torre, P. Rad, and K.-K. R. Choo, "Driverless vehicle security: Challenges and future research opportunities," *Future Gener. Comput. Syst.*, vol. 108, pp. 1092–1111, 2020.

[10] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, Feb. 2019.

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

[12] M. Hashem Eiza and Q. Ni, "Driving with sharks: Rethinking connected vehicles with vehicle cybersecurity," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 45–51, Jun. 2017.

[13] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social attribute aware incentive mechanism for device-to-device video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1908–1920, Aug. 2017.

[14] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9390–9401, Sep. 2020.

[15] K. Fan, W. Jiang, Q. Luo, H. Li, and Y. Yang, "Cloud-based RFID mutual authentication scheme for efficient privacy preserving in IoV," *J. Franklin Inst.*, vol. 358, no. 1, pp. 193–209, Jan. 2021.

[16] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1654–1667, 2020.

[17] M. Wazid, A. K. Das, V. Bhat, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," *J. Netw. Comput. Appl.*, vol. 150, 2020, Art. no. 102496.

[18] J. Shen, D. Liu, X. Chen, J. Li, N. Kumar, and P. Vijayakumar, "Secure real-time traffic data aggregation with batch verification for vehicular cloud in VANETs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 807–817, Jan. 2020.
[19] L. Cui et al., "A blockchain-based containerized edge computing platform for the Internet of Vehicles," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2395–2408, Feb. 2021.
[20] Y. Liu, L. Wang, and H. Chen, "Message authentication using proxy vehicles in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 8, pp. 3697–3710, Aug. 2015.
[21] L. Song, G. Sun, H. Yu, X. Du, and M. Guizani, "FBIA: A fog-based identity authentication scheme for privacy preservation in Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5403–5415, May 2020.
[22] H. Liu, H. Wang, and H. Gu, "HPBS: A hybrid proxy based authentication scheme in VANETs," *IEEE Access*, vol. 8, pp. 161655–161667, 2020.
[23] A. K. Sutrala, P. Bagga, A. K. Das, N. Kumar, J. J. P. C. Rodrigues, and P. Lorenz, "On the design of conditional privacy preserving batch verification-based authentication scheme for Internet of Vehicles deployment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5535–5548, May 2020.
[24] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.
[25] Z. Xiong, J. Kang, D. Niyato, P. Wang, and H. V. Poor, "Cloud/edge computing service management in blockchain networks: Multi-leader multi-follower game-based ADMM for pricing," *IEEE Trans. Serv. Comput.*, vol. 13, no. 2, pp. 356–367, Mar./Apr. 2020.
[26] X. Wang, P. Zeng, N. Patterson, F. Jiang, and R. Doss, "An improved authentication scheme for Internet of Vehicles based on blockchain technology," *IEEE Access*, vol. 7, pp. 45061–45072, 2019.
[27] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An efficient decentralized key management mechanism for VANET with blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5836–5849, Jun. 2020.
[28] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan, and Y. Zhang, "Blockchain empowered cooperative authentication with data traceability in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4221–4232, Apr. 2020.
[29] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, "Privacy-preserving cloud-based road condition monitoring with source authentication in VANETs," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 7, pp. 1779–1790, Jul. 2019.
[30] J. Zhang, H. Zhong, J. Cui, M. Tian, Y. Xu, and L. Liu, "Edge computing-based privacy preserving authentication framework and protocol for 5G-enabled vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7940–7954, Jul. 2020.
[31] R. Sharma and S. Chakraborty, "Blockapp: Using blockchain for authentication and privacy preservation in IoV," in *Proc. IEEE Globecom Workshops*, 2018, pp. 1–6.
[32] D. Gabay, K. Akkaya, and M. Cebe, "Privacy-preserving authentication scheme for connected electric vehicles using blockchain and zero knowledge proofs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5760–5772, Jun. 2020.
[33] C. Guo et al., "Pingmesh: A large-scale system for data center network latency measurement and analysis," in *Proc. ACM Conf. Special Int. Group Data Commun.*, 2015, pp. 139–152.
[34] S. K. Barker and P. Shenoy, "Empirical evaluation of latency-sensitive application performance in the cloud," in *Proc. 1st Annu. ACM SIGMM Conf. Multimedia Syst.*, 2010, pp. 35–46.
[35] T. Yasui, Y. Ishibashi, and T. Ikedo, "Influences of network latency and packet loss on consistency in networked racing games," in *Proc. 4th ACM SIGCOMM Workshop Netw. Syst. Support Games*, 2005, pp. 1–8.
[36] N. Chowdhary and P. Deep Kaur, "Addressing the characteristics of mobility models in IoV for smart city," in *Proc. Int. Conf. Comput., Commun. Automat.*, 2016, pp. 1298–1303.
[37] D. Mahto and D. K. Yadav, "RSA and ECC: A comparative analysis," *Int. J. Appl. Eng. Res.*, vol. 12, no. 19, pp. 9053–9061, 2017.
[38] J. H. Silverman and J. Suzuki, "Elliptic curve discrete logarithms and the index calculus," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 1998, pp. 110–125.
[39] J. H. Cheon, "Security analysis of the strong Diffie-Hellman problem," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2006, pp. 1–11.
[40] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Bus. Inf. Syst. Eng.*, vol. 59, no. 3, pp. 183–187, 2017.
[41] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Tang, "Secure SVM training over vertically-partitioned datasets using consortium blockchain for vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5773–5783, Jun. 2020.
[42] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Serv.*, vol. 14, no. 4, pp. 352–375, 2018.
[43] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," *Financial Innov.*, vol. 2, no. 1, 2016, Art. no. 24.
[44] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.
[45] W.-J. Chang, L.-B. Chen, and K.-Y. Su, "DeepCrash: A deep learning-based internet of vehicles system for head-on and single-vehicle accident detection with emergency notification," *IEEE Access*, vol. 7, pp. 148163–148175, 2019.
[46] C. Li et al., "Transparent AAA security design for low-latency MEC-integrated cellular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3231–3243, Mar. 2020.
[47] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4585–4600, Jun. 2019.
[48] Q. Ding, H. Pang, and L. Sun, "Location dependent pricing in edge caching market with non-uniform popularity," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–7.
[49] H. Arshad, M. A. Shah, H. A. Khattak, Z. Ameer, A. Abbas, and S. U. Khan, "Evaluating bio-inspired optimization techniques for utility price estimation in fog computing," in *Proc. IEEE Int. Conf. Smart Cloud*, 2018, pp. 84–89.
[50] M. Shen, H. Liu, L. Zhu, K. Xu, and M. Guizani, "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 942–954, May 2020.
[51] J. Xu, K. Xue, H. Tian, J. Hong, D. S. L. Wei, and P. Hong, "An identity management and authentication scheme based on redactable blockchain for mobile networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6688–6698, Jun. 2020.
[52] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, Jul. 2017.
[53] Wikipedia contributors, "Speed limits by country – Wikipedia, the free encyclopedia," Accessed on: Nov. 25, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Speed_limits_by_country&oldid=989881417
[54] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–15.
[55] Y. Xu et al., "Artificial intelligence: A powerful paradigm for scientific research," *Innovation*, vol. 2, no. 4, 2021, Art. no. 100179.

Meng Shen (Member, IEEE) received the B.Eng. degree in computer science from Shandong University, Jinan, China, in 2009, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2014. He is currently a Professor with the Beijing Institute of Technology, Beijing, China. He has authored more than 50 papers in top-level journals and conferences, such as *ACM SIGCOMM*, *IEEE Journal on Selected Areas in Communications*, and *IEEE Transactions on Information Forensics and Security*. His research interests include data privacy and security, blockchain applications, and encrypted traffic classification. He has guest edited special issues on emerging technologies for data security and privacy in *IEEE Network* and *IEEE Internet of Things Journal*. He was the recipient of the Best Paper Runner-Up Award at IEEE IPCCC 2014 and IEEE/ACM IWQoS 2020. Dr. Shen was selected by the Beijing Nova Program 2020 and was the winner of the ACM SIGCOMM China Rising Star Award 2019.

Hao Lu received the B.Eng. degree in computer science and technology from Shandong University, Jinan, China, in 2021. She is currently working toward the master's degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. Her research interests on cyberspace security.

Fei Wang (Member, IEEE) born in 1988. He received the B.S. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2011, and the Ph.D. degree in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2017. He is currently a Ph.D. Associate Professor. From 2017 to 2020, he was a Research Assistant with the Institute of Computing Technology, Chinese Academy of Sciences. Since 2020, he has been an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences. His main research interests include spatiotemporal data mining, fusion, graph, and neural networks.

Huisen Liu received the B.Eng. degree in computer science from Northwest Agriculture and Forestry University of China, Xianyang, China, in 2018 and the master's degree from the Beijing Institute of Technology, Beijing, China, in 2021. His research interests include blockchain applications and IoT security.

Liehuang Zhu (Senior Member, IEEE) is currently a Professor with the Department of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China. He is selected into the Program for New Century Excellent Talents in University from the Ministry of Education, China. His research interests include Internet of Things, cloud computing security, internet and mobile security.