Fortifying Smart Transportation Security Through Public Blockchain

Mohammad Wazid, *Senior Member, IEEE*, Basudeb Bera, Ashok Kumar Das, *Senior Member, IEEE*, Saraju P. Mohanty, *Senior Member, IEEE*, and Minho Jo, *Senior Member, IEEE*

***Abstract*—Smart vehicles-enabled intelligent transportation system (ITS) supports a wide range of applications, such as, but not limited to, traffic planning and management, collision avoidance alert system, automated road speed enforcement, electronic toll collection, and real-time parking management, to name a few. However, it suffers from various types of security and privacy issues due to insecure communication among the entities over public channels. Therefore, an efficient and lightweight security mechanism is essential to protect the data that is both at rest as well as in transit. To this direction, we propose a public blockchain-envisioned secure communication framework for ITS (PBSCF-ITS). The proposed PBSCF-ITS guarantees access control and key management among the vehicle to vehicle, vehicle to roadside unit, and roadside unit to cloud server. We analyze the security of PBSCF-ITS to prove its resilience against various types of possible attacks. Furthermore, the performance of PBSCF-ITS with other related competing schemes has been compared. The obtained results illustrate that PBSCF-ITS outperforms the existing ones. Additionally, the pragmatic study of PBSCF-ITS is conducted to check its influence on various network-related performance parameters, like the number of mined blocks and transactions per block.**

***Index Terms*—Access control and key management, blockchain, intelligent transportation system (ITS), security, vehicular network.**

I. Introduction

An INTELLIGENT transportation system (ITS) is a technological platform that has the capability of sensing, analysis, control, and communication to enable safe, reliable, and infotainment-enabled experience for commuters. It enables safe and secure and infotainment-rich driving experience by keeping the cyber-attackers at the bay from attacking ITS and improving the driving experience [1]–[3]. ITS is realized through vehicular networks and consists of smart vehicles, roadside units (RSUs), sensing units, environmental monitoring system, traffic monitoring, and surveillance system [4]–[6]. Vehicular networks use different communication technologies, including the dedicated short-range communication (DSRC), Bluetooth, WiFi, and cellular networks [7]. These technologies enable different modes of communication, such as Vehicle to Vehicle (V2V) and Vehicle to Everything (V2X) (that includes Vehicle-to-Cloud communication). Moreover, it produces massive amount of data (referred to as Big traffic data) that needs to be stored, processed, and analyzed in a secure way. The conducted analysis on this data is further helpful in predicting the important factors in transportation, such as chances of roadside accident, environmental conditions, driver behavior, expected travel time, and congestion on a specific route, to name a few [8], [9].

Due to the increased number of vehicles pervading the roads, the realization of ITS is essential because the ever-growing traffic surpasses the capacity of the existing infrastructure. However, such system warrants the deployment of secure data management and sharing techniques (for both data at rest and in transit) [2], [8]. Here, the mechanism of blockchain can play an important role as it is tamper proof, decentralized, anonymous, and robust against various types of information security-related attacks [10]–[12]. Therefore, the use of the blockchain mechanism is strongly suggested to introduce for such kind of communication environment [13]. It is worth mentioning that vehicular networks use different communication technologies that enable different modes of communication, such as V2V, Vehicle to RSU (V2RSU), and RSU to cloud (RSU2C) [14], [15].

There are other applications that use the blockchain mechanism. A decoupled blockchain-based approach for the edge-envisioned ecosystem was presented by Aujla and Jindal [16]. This approach used the nearby edge devices in order to create the decoupled blocks into the blockchain. This can provide the secure exchange of healthcare data from sensors to the edge nodes [17]. The real-time processing is needed for energy trading computation, which is an important requirement of some computing environments, like Tactile Internet. Therefore, to address such challenges, a blockchain-based secure energy trading scheme for electric vehicles (EVs) was presented by Chaudhary *et al.* [18]. This scheme also ensures resilience against the single point of failure.

Manuscript received 14 October 2021; revised 4 January 2022; accepted 7 February 2022. Date of publication 11 February 2022; date of current version 24 August 2022. This work was supported in part by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE), South Korea, under Grant 2021RIS-004, and in part by the Ripple Centre of Excellence Scheme, CoE in Blockchain (Sanction No. IIIT/Research and Development Office/Internal Projects/001/2019), IIIT Hyderabad, India. *(Corresponding author: Minho Jo.)*

Mohammad Wazid is with the Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India (e-mail: wazidkec2005@gmail.com).

Basudeb Bera and Ashok Kumar Das are with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad 500 032, India (e-mail: basudeb.bera@research.iiit.ac.in; iitkgp.akdas@gmail.com).

Saraju P. Mohanty is with the Department of Computer Science and Engineering, University of North Texas, Denton, TX 76203 USA (e-mail: saraju.mohanty@unt.edu).

Minho Jo is with the Department of Computer Convergence Software, Korea University, Sejong 30019, South Korea (e-mail: minhojo@korea.ac.kr).

Digital Object Identifier 10.1109/JIOT.2022.3150842

We arrange the sections of this article in the following way. The motivation and novel contributions of this article are given in Section II. The literature study of related prior works is given in Section III. The network model and adversary model associated with the proposed public blockchain-envisioned secure communication framework for ITS (PBSCF-ITS) are provided in Section IV. The different phases of the proposed PBSCF-ITS are elaborated in Section V. The essential security analysis of the proposed PBSCF-ITS is provided in Section VI. A rigorous comparative study among PBSCF-ITS and other relevant schemes is stated in Section VIII. The practical implementation of PBSCF-ITS is specified in Section VII. Finally, the work is concluded in Section IX.

II. Motivation and Research Contributions

The motivation and novel contributions of this article are provided as follows.

A. Motivation

Smart vehicles-enabled ITS supports and provides a broad range of applications and services. However, communication in such an environment has security and privacy issues, and different attacks can be launched to either tamper with the data or disrupt the normal communication. The communication among the vehicles, RSUs, and cloud servers (CSs) takes place through wireless medium which is prone to a myriad of cyber-threats. For instance, an adversary may tamper with the communicated information among different parties in such a communication environment. Different potential attacks in this environment include "replay," "Man-in-The-Middle (MiTM)," "impersonation," "illegal session key communication," "credential leakage," and other forms of data disclosure attacks. The front line of defense against most of such attacks is an effective and robust access control and key establishment mechanism. Through such a mechanism, the entities, such as vehicles, RSUs, and CSs, can authenticate with each other and can then establish session keys for their secure communication. Moreover, the blockchain mechanism is essential for such kind of communication environment, because it is tamper-proof, decentralized, anonymous, and robust against various types of information security-related attacks [19]. Therefore, it is imperative to provide a blockchain-based access control and key establishment mechanism for the smart vehicles-enabled ITS communication [15], [20]–[24]. Thus, we design a new a PBSCF-ITS by having an access control and key establishment scheme, where "vehicle-to-vehicle," "vehicle-to-RSU" and "RSU-to-CS" session key establishments take place. These processes will help the entities to exchange their data in a secure way.

B. Research Contributions

Our contributions in this article are listed as follows.

1) We design the network and adversary models for the smart vehicles-enabled ITS.
2) We propose a PBSCF-ITS. The blockchain technology makes such a designed framework more secure, reliable, and decentralize. The smart transportation security is fortified through the public blockchain.
3) PBSCF-ITS allows access control and key management among V2V, V2RSU, and RSU2C at the same time.
4) A rigorous security analysis and a detailed comparative study among the proposed PBSCF-ITS and other existing state-of-the-art schemes show that the performance of PBSCF-ITS is better than existing schemes in terms of superior security and more functionality features, and low or comparable communication/computational overheads.
5) The pragmatic blockchain-based simulation study of PBSCF-ITS shows its influence on the performance parameters, such as computational time (seconds) versus "number of mined blocks" and "transactions per block," and "transactions per second" versus "number of mined blocks."

III. Related Prior Works

To date, there has been a number of papers that address authentication, access control, and key management in ITS.

A survey on the history and characteristics of big data and its role in ITS was conducted by Zhu *et al.* [1]. Furthermore, they also presented a framework for big data analytics in ITS. Several case studies of big data analytics applications in ITS, such as "road traffic accidents analysis," "road traffic flow prediction," "public transportation service plan," "rail transportation management and control," etc., were also discussed. In another work, Pribyl *et al.* [2] proposed a smart city model based on ITS communication. Furthermore, some guidance for the establishment of the smart city architecture to overcome the system complexity was also provided.

Herrera-Quintero *et al.* [3] designed an ITS smart sensor prototype by incorporating the Internet of Things (IoT) and using the "Serverless and Microservice Architecture" for the planning of transportation system utilized in []bus rapid transit (BRT) systems. Similarly, Kaffash *et al.* [8] conducted a comprehensive review of the applications of ITS. They also provided a review of most of the recognized models with big data applicable in the ITS context. Lian *et al.* [9] reviewed some studies which used big data to analyze the traffic safety in ITS and connected/automated vehicles (CAVs) communication environment. The focus was on topics, such as crash prediction and detection and the factors, which contributed to the crash, driving behavior, and so on.

Wazid *et al.* [15] proposed an authentication and key management scheme to secure the communication among vehicles, RSUs, fog, and CSs in the fog computing-based Internet of Vehicles (IoV) communication paradigm. Later on, Vangala *et al.* [20] proposed a blockchain-endowed authentication mechanism that is based on digital certificates to detect vehicular accidents and disseminate notification in ITS. In their scheme, each vehicle securely notifies the accident-related information to its adjacent cluster head (CH) in case of any accident. Similarly, Liu *et al.* [21] proposed an authentication mechanism for IoV communication. They used mostly focused on security and privacy preservation

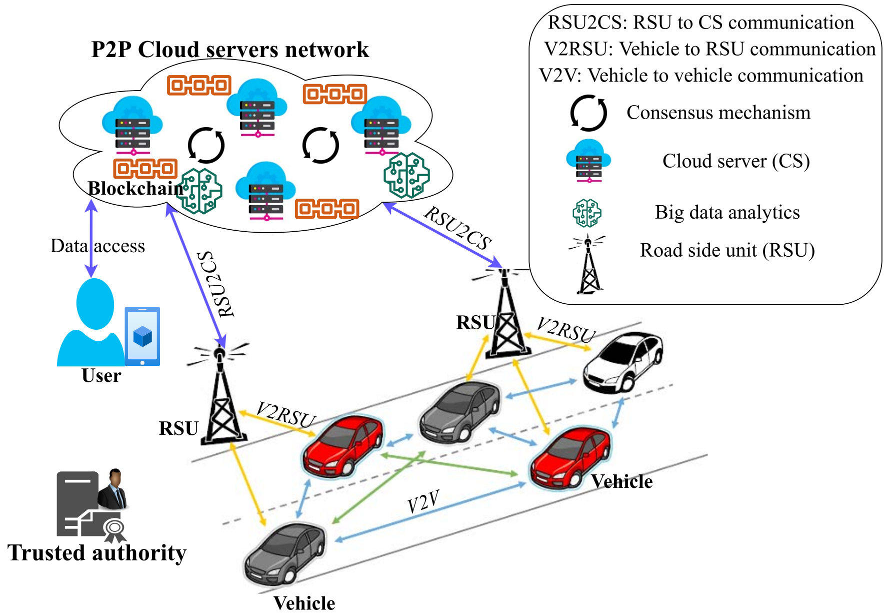


Fig. 1. Network model (adapted from [15] and [20]).

through a dual authentication method for IoV communication. Egala *et al.* [25] presented a hybrid computing mechanism with blockchain-based distributed data storage system (DDSS) to overcome the drawbacks (i.e., high delay, storage cost, and single point of failure) of blockchain-based cloud-centric IoMT healthcare system. Biswas *et al.* [26] presented a lightweight proof of block and trade consensus mechanism for IoT blockchain along with an integration framework. The provided mechanism allowed the validation of trades as well as blocks with less computation cost.

In another work, a mechanism for secure communication between the vehicles and RSUs through a certificateless short signature (CLSS) method was presented by Liu *et al.* [27]. The unforgeability property of their scheme was also proven through a random oracle model. On the other hand, Cui *et al.* [22] proposed RSU-based authentication and the dissemination of authentication information to nearby vehicles to improve the efficiency of authentication. In their scheme, an RSU can authenticate vehicles and also broadcasts the authentication results to the nearby vehicles to reduce unnecessary authentication and raise the efficiency of the communication system.

Pokhrel *et al.* [28] designed a "privacy-aware automated parking model for smart autonomous vehicles." Their model is based on both differential privacy and zero-knowledge proof, where location privacy and identity privacy are addressed. Specifically, their model is able to resist multiple reservation attacks intended by the illegal users. Moreover, their model can protect user location privacy by means of applying the differential privacy schemes.

In vehicular cyber–physical systems (VCPSs), both computing and physical resources are integrated in order to interact among each other as well as their nearby environment in order to improve the safety, efficiency, and infotainment quality associated with the transportation. Lu *et al.* [29] suggested a scheme that can handle to mitigate data leakage in VCPS, which is based on federated learning. They also designed a random subgossip updating scheme for protecting the privacy during the learning procedure.

IV. Model of the Proposed System

This section talks about the network and adversarial models for the proposed PBSCF-ITS.

A. Network Model

The network model of PBSCF-ITS is given in Fig. 1 that consists of smart vehicles, RSUs, CSs, users, and traffic monitoring and surveillance system. A smart vehicle can communicate with other nearby smart vehicles or RSU through DSRC or cellular networks whereas vehicles communicate with CSs through cellular communication networks. Furthermore, RSU can communicate with the back-end systems (such as cloud or registration authorities) through either wired or wireless networks. However, the communication between smart vehicle and CS may happen through some wireless communication technology such as a cellular network. Similarly, RSU can communicate with the CS through back-end communication, for instance, either wired or wireless backbone communication. The traffic monitoring and surveillance system is connected to the CS through back-end communication, like wired or wireless backbone communication. The sensing and monitoring systems in vehicles sense the data from their surroundings and send the information to the CSs for additional processing and storage. Other network entities also generate data and send it to the CS. Thus, in ITS, an enormous amount of data is generated by different sources and therefore termed as Big traffic data. We need some Big data analytics methods, which enable us to acquire useful information, such as prediction on road and environmental condition, driver behavior, and traffic condition.

The data of the ITS environment is stored in the form of a public blockchain over the peer-to-peer CS (P2PCS) network. The use of the blockchain provides protection against some potential attacks, like the data disclosure attack and data modification attack. According to the discussed network model, the following types of secure communications take place: V2V, V2RSU, and RSU2C communication, traffic monitoring and surveillance (CCTV) system to CS communication, and user to CS (U2C) communication. The entire communication happens through some wireless or wired communication technology. However, such type of communication is open to the network attackers and it can be compromised through different types of attacks as discussed earlier. The openness of wireless channel in vehicular networks inherently lure attackers to launch different attacks (discussed in the adversary model). Therefore, the use of secure blockchain-based access control and key establishment scheme seems essential. Hence, to protect the communication, a secure public blockchain-based access control and key establishment scheme has been designed.

B. Adversary Model

We use the widely used Dolev–Yao (DY) adversary model for the proposed PBSCF-ITS. According to the DY model, the communicating entities communicate over a public medium which is prone to eavesdropping and other cyber attacks. The end point entities, such as smart vehicles, RSUs, and end users, are not generally untrustworthy. Therefore, the communicated messages may be delayed, updated, dropped, or modified. Moreover, the CS is assumed to be semitrusted entity in the ITS environment and the trusted authority (TA), responsible for entity registration, is considered as the fully trusted entity of the network. Furthermore, we also follow the guidelines of "Canetti and Krawczyk's (CK) adversary model [30]" that is more powerful model than the DY model and can be utilized in authentication, access control, and key establishment mechanisms. According to the "CK-adversary model," an adversary $\mathcal{A}$ enjoys all the facilities that are provided under the DY model including extra capabilities, such as compromise of secret credentials via session-hijacking attacks. There is also a chance that $\mathcal{A}$ may steal some of the onboard units OBUs of some smart vehicles as in sensor nodes [31], and later may try to acquire sensitive information from its memory with the help of advanced power analysis attacks [32]. The acquired information can be then made use of launching other attacks, such as impersonation and illegal session key computation attacks.

V. Proposed Blockchain-Based Framework

In this section, we explain in detail the proposed PBSCF-ITS. After the execution of all steps of PBSCF-ITS, there will be the access control (to access data among vehicles) and key management between a vehicle and the other vehicles, vehicle to the RSU, vehicle to the CS, and RSU to the CS. The inclusion of the blockchain makes this framework more secure, reliable, and decentralize, which are the essential requirements of an ITS. PBSCF-ITS is divided into the following phases: 1) system initialization; 2) registration, access control, and key establishment; 3) dynamic smart vehicle addition; and 4) block creation, verification, and addition phase, which are discussed as follows.

To achieve protection against strong replay attack, we assume that the clocks of the communicating entities in the network are synchronized, which is a normal supposition utilized in designing various networking environments related to authentication protocols [15], [20], [33]–[36].

A. System Initialization Phase

In the system initialization phase, some important cryptographic primitives and parameters are selected that are needed for other phases, such as "registration, access control, and key agreement." A TA selects a "nonsingular elliptic curve over a finite field" by picking two constants $u \in Z_q$ and $v \in Z_q$, where $Z_q = \{0, 1, \dots, q-1\}$ and $q > 3$ be a prime number such that "$4u^3 + 27v^2 \neq 0 \pmod q$," of the form: "$y^2 = x^3 + ux + v$ over $GF(q)$" having $\mathcal{O}$ as a point at infinity or zero point. Suppose G is taken as a base point in $E_q(u, v)$ having an order as big as q. Furthermore, TA selects a "one-way (collision-resistant) hash function $h(\cdot)$ (for instance, SHA-256 hashing algorithm [37])."

B. Registration Phase

The participating entities must be registered before using the network services. The TA performs registration of various entities in the offline mode through a secure channel. The registration of different network entities is discussed as follows.

1) Registration of Smart Vehicles: The TA uses the following steps to register a smart vehicle, say V_i.

RV1: First, TA generates its own private key $s_{\text{TA}} \in Z_q^* = \{1, 2, \dots, q-1\}$ and computes the respective public key as $Q_{\text{TA}} = s_{\text{TA}}.G$, where $x.G$ is the point multiplication on the specified elliptic curve and $x \in Z_q^*$. Then, TA generates a private key of smart vehicle V_i as $s_{V_i} \in Z_q^*$ and calculates the corresponding public key as $Q_{V_i} = s_{V_i}.G$.

RV2: TA selects ID_{V_i} and ID_{TA} as the identities of V_i and itself, respectively, and calculates the corresponding pseudoidentity of V_i as $\text{RID}_{V_i} = h(\text{ID}_{V_i}||s_{\text{TA}})$ and its own pseudoidentity as $\text{RID}_{\text{TA}} = h(\text{ID}_{\text{TA}}||s_{\text{TA}})$. TA also computes the temporal credential of V_i as $\text{TC}_{V_i} = h(\text{ID}_{V_i} \; || \; \text{RTS}_{V_i} \; || \; s_{V_i} \; || \; s_{\text{TA}}||\text{RID}_{\text{TA}})$, where RTS_{V_i} is the registration timestamp of V_i. In addition, TA generates a random secret $n_{V_i} \in Z_q^*$ to compute its corresponding public parameter $N_{V_i} = n_{V_i}.G$.

RV3: TA generates the certificate for V_i as $\text{CT}_{V_i} = s_{\text{TA}} + h(Q_{\text{TA}} \; || \; Q_{V_i}) * n_{V_i} \pmod q$, where $*$ represents a modular multiplication in Z_q^*. Note that $n_{V_i} \in Z_q^*$ is different for different vehicles, and TA announces N_{V_i} publicly.

RV4: TA finally stores the credentials $\{\text{RID}_{V_i}, \text{TC}_{V_i}, (s_{V_i}, Q_{V_i}), \text{CT}_{V_i}, h(\cdot), E_q(u, v), G\}$ in the onboard unit OBU_{V_i} of V_i before its deployment. To protect against potential attacks, TA deletes sensitive parameters, such as n_{V_i} and RTS_{V_i} from its database and makes the declaration of the public parameters

Registration of smart vehicle V_i	
Trusted authority (TA)	Smart vehicle (V_i)
Generate $s_{TA} \in Z_q^*$. Compute $Q_{TA} = s_{TA}.G$. Generate $s_{V_i} \in Z_q^*$. Compute $Q_{V_i} = s_{V_i}.G$. Select ID_{V_i} & ID_{TA}. Compute $RID_{V_i} = h(ID_{V_i}\|\|s_{TA})$, $RID_{TA} = h(ID_{TA}\|\|s_{TA})$, $TC_{V_i} = h(ID_{V_i}\ \|\|RTS_{V_i}\|\|\ s_{V_i}$ $\|\|s_{TA}\|\|RID_{TA})$, Generate $n_{V_i} \in Z_q^*$. Compute $N_{V_i} = n_{V_i}.G$, $CT_{V_i} = s_{TA} + h(Q_{TA}\ \|\|Q_{V_i}) * n_{V_i}$ $\pmod q$. Store $\{RID_{V_i}, TC_{V_i}, (s_{V_i}, Q_{V_i})$, $CT_{V_i}, h(\cdot), E_q(u,v), G\}$ in OBU_{V_i}	
	V_i is deployed with OBU_{V_i} with credentials $\{RID_{V_i}, TC_{V_i}$, $(s_{V_i}, Q_{V_i}), CT_{V_i}$, $h(\cdot), E_q(u,v), G,\}$.

Fig. 2. Registration of a smart vehicle V_i.

publicly. The summary of the registration process of smart vehicle V_i is given in Fig. 2.

2) Registration of RSU: The TA uses the following steps to register an RSU, say RSU_l.

RRSU1: TA first generates a private key for RSU_l as $s_{\text{RSU}_l} \in Z_q^*$ and derives the respective public key as $Q_{\text{RSU}_l} = s_{\text{RSU}_l}.G$.

RRSU2: TA selects ID_{RSU_l} as the identity of RSU_l and calculates the corresponding pseudoidentity of RSU_l as $\text{RID}_{\text{RSU}_l} = h(\text{ID}_{\text{RSU}_l}||s_{\text{TA}})$. TA also computes the temporal credential of RSU_l as $\text{TC}_{\text{RSU}_l} = h(\text{ID}_{\text{RSU}_l}\ ||\ \text{RTS}_{\text{RSU}_l}\ ||\ s_{\text{RSU}_l}\ ||\ s_{\text{TA}}||\text{RID}_{\text{TA}})$, where $\text{RTS}_{\text{RSU}_l}$ is the registration timestamp of RSU_l. Furthermore, TA picks a random secret $n_{\text{RSU}_l} \in Z_q^*$ to compute its corresponding public parameter $N_{\text{RSU}_l} = n_{\text{RSU}_l}.G$.

RRSU3: TA calculates the certificate of RSU_l as $\text{CT}_{\text{RSU}_l} = s_{\text{TA}} + h(Q_{\text{TA}}\ ||\ Q_{\text{RSU}_l}) * n_{\text{RSU}_l} \pmod q$. Note that the random secret $n_{\text{RSU}_l} \in Z_q^*$ is different for the RSUs. Further, the TA announces N_{RSU_l} publicly.

RRSU4: TA stores the credentials $\{\text{RID}_{\text{RSU}_l}, \text{TC}_{\text{RSU}_l}, (s_{\text{RSU}_l}, Q_{\text{RSU}_l}), \text{CT}_{\text{RSU}_l}, h(\cdot), E_q(u,v), G\}$ in RSU_l's memory before its stationing. TA deletes sensitive values, such as n_{RSU_l} and $\text{RTS}_{\text{RSU}_l}$ from its database to overcome the security issues. TA publicly makes the declaration of all public parameters. The summary of the registration process of roadside unit RSU_l is given in Fig. 3.

3) Registration of Cloud Servers: The TA also carries out the registration of a CS CS_k using the following steps.

RCS1: TA first generates a private key of CS_k as $s_{\text{CS}_k} \in Z_q^*$ to calculate the corresponding public key as $Q_{\text{CS}_k} = s_{\text{CS}_k}.G$. Again, TA selects CS_k's identity as ID_{CS_k} and calculates the corresponding pseudoidentity as $\text{RID}_{\text{CS}_k} = h(\text{ID}_{\text{CS}_k}||s_{\text{TA}})$ and the temporal credential of CS_k as $\text{TC}_{\text{CS}_k} = h(\text{ID}_{\text{CS}_k}\ ||\ \text{RTS}_{\text{CS}_k}\ ||\ s_{\text{CS}_k}\ ||\ s_{\text{TA}}\ ||\ \text{RID}_{\text{TA}})$, where RTS_{CS_k} is the CS_k's registration timestamp.

RCS2: TA sends the credentials RID_{CS_k}, TC_{CS_k}, $(s_{\text{CS}_k}, Q_{\text{CS}_k})$ to CS_k through a secure channel using a shared key K_{TA,CS_k} between them. In addition, TA also provides the registration information of the vehicles and RSUs

Registration of road side unit RSU_l	
Trusted authority (TA)	RSU (RSU_l)
Generate $s_{RSU_l} \in Z_q^*$. Compute $Q_{RSU_l} = s_{RSU_l}.G$. Select ID_{RSU_l} and calculate $RID_{RSU_l} = h(ID_{RSU_l}\|\|s_{TA})$, $TC_{RSU_l} = h(ID_{RSU_l}\ \|\|RTS_{RSU_l}$ $\|\|s_{RSU_l}\ \|\|s_{TA}\|\|RID_{TA})$. Select $n_{RSU_l} \in Z_q^*$ and compute $N_{RSU_l} = n_{RSU_l}.G$, $CT_{RSU_l} = s_{TA} + h(Q_{TA}$ $\|\|Q_{RSU_l}) * n_{RSU_l} \pmod q$. Store $\{RID_{RSU_l}, TC_{RSU_l}$, (s_{RSU_l}, Q_{RSU_l}), $h(\cdot), E_q(u,v), G\}$ in RSU_l.	
	RSU_l is deployed with $\{RID_{RSU_l}$, $TC_{RSU_l}, (s_{RSU_l}, Q_{RSU_l})$, $CT_{RSU_l}, h(\cdot), E_q(u,v), G\}$

Fig. 3. Registration of roadside unit RSU_l.

Registration of cloud server CS_k	
Trusted authority (TA)	Cloud server (CS_k)
Generate $s_{CS_k} \in Z_q^*$. Compute $Q_{CS_k} = s_{CS_k}.G$. Select ID_{CS_k} and compute $RID_{CS_k} = h(ID_{CS_k}\|\|s_{TA})$, $TC_{CS_k} = h(ID_{CS_k}\ \|\|RTS_{CS_k}$ $\|\|s_{CS_k}\ \|\|s_{TA}\ \|\|RID_{TA})$. $\xrightarrow{\{RID_{CS_k}, TC_{CS_k}, (s_{CS_k}, Q_{CS_k})\}}$ (through secure channel)	
	CS_k is deployed with credentials $\{RID_{CS_k}, TC_{CS_k}, (s_{CS_k}, Q_{CS_k})$, $E_q(u,v), G, h(\cdot)\}$.

Fig. 4. Registration of cloud server CS_k.

that are located in that particular region to its corresponding CS CS_k through secure channel.

RCS3: After receiving the registration parameters from TA, CS_k stores the credentials $\{\text{RID}_{\text{CS}_k}, \text{TC}_{\text{CS}_k}, (s_{\text{CS}_k}, Q_{\text{CS}_k}), E_q(u,v), G, h(\cdot)\}$ in its secure database. CS_k publicizes its public parameters. The summary of the registration process of CS CS_k is also given in Fig. 4.

Remark 1: Note that the TA deletes all secret information, like the private keys and registration timestamp values from its own memory. Therefore, it is not feasible for the adversary (including the privileged-insider user) to execute potential attacks, such as "privileged-insider attack," "unauthorized session key computation attack," and "impersonation attack." Apart from that, RSU_l and CS_k store all their secret data in the secure region of their memory for the protection of stolen verifier attack and other associated attacks.

C. Access Control Phase

This phase is required to provide secure access control among different smart vehicles, and vehicle and its nearby RSU. In this phase, we consider that a vehicle (V_i) can establish a secure connection with its associated cluster head (V_j) to share data directly among them. Moreover, the access control can also be performed between a vehicle V_i and its related RSU_l. Both types of mechanisms are discussed as follows.

1) Access Control Between Vehicles V_i and V_j: We need to execute the following steps to perform this task.

ACVV1: V_i initiates the access control process by generating a random secret $r_{V_i} \in Z_q^*$ and a current timestamp T_1, and then computing $A_{V_i} = h(\mathrm{RID}_{V_i} \mid\mid \mathrm{TC}_{V_i} \mid\mid r_{V_i} \mid\mid T_1)$, $R_{V_i} = r_{V_i} \cdot G$, $\mathrm{CT}^*_{V_i} = \mathrm{CT}_{V_i} \oplus h(r_{V_i} \cdot Q_{V_j} \mid\mid T_1)$, $M_1 = A_{V_i} \oplus h(s_{V_i} \cdot Q_{V_j} \mid\mid R_{V_i} \mid\mid T_1)$ and the ElGamal-type signature as $M_2 = s_{V_i} + h(M_1 \mid\mid \mathrm{CT}^*_{V_i} \mid\mid Q_{\mathrm{TA}} \mid\mid Q_{V_i}) * r_{V_i} \pmod q$. After the calculations of these parameters, V_i sends the message $\mathrm{Msg}_1 = \{M_1, M_2, R_{V_i}, \mathrm{CT}^*_{V_i}, T_1\}$ to V_j through public channel.

ACVV2: Upon the arrival of Msg_1 from V_i at time T_1^*, V_j first proceeds for the verification of timeliness of T_1 through the condition: $|T_1 - T_1^*| \leq \Delta T$, given the "maximum transmission delay is ΔT." If it matches, it then verifies the signature as $M_2 \cdot G = Q_{V_i} + h(M_1 \mid\mid \mathrm{CT}^*_{V_i} \mid\mid Q_{\mathrm{TA}} \mid\mid Q_{V_i}) \cdot R_{V_i}$. If it is successfully verified, the next step is followed.

ACVV3: V_j proceeds for the generation of a random secret $r_{V_j} \in Z_q^*$ along with a fresh timestamp value T_2. Next, it derives $\mathrm{CT}_{V_i} = \mathrm{CT}^*_{V_i} \oplus h(r_{V_j} \cdot Q_{V_i} \mid\mid T_1)$ and verifies the certificate by $\mathrm{CT}_{V_i} \cdot G = Q_{\mathrm{TA}} + h(Q_{\mathrm{TA}} \mid\mid Q_{V_i}) \cdot N_{V_i}$. After successfully validation, V_j computes $A_{V_j} = h(\mathrm{RID}_{V_j} \mid\mid \mathrm{TC}_{V_j} \mid\mid r_{V_j} \mid\mid T_2)$, $M_3 = A_{V_j} \oplus h(s_{V_j} \cdot Q_{V_i} \mid\mid \mathrm{CT}_{V_j} \mid\mid T_1)$, $\mathrm{CT}^*_{V_j} = \mathrm{CT}_{V_j} \oplus h(r_{V_j} \cdot Q_{V_i} \mid\mid T_1)$, and $A_{V_i} = M_1 \oplus h(s_{V_j} \cdot Q_{V_i} \mid\mid R_{V_i} \mid\mid T_1)$. After that, V_j calculates a session key as $\mathrm{SK}_{V_j,V_i} = h(A_{V_i} \mid\mid A_{V_j} \mid\mid \mathrm{CT}_{V_i} \mid\mid \mathrm{CT}_{V_j} \mid\mid T_1 \mid\mid T_2)$, and session key verifier by $M_4 = h(\mathrm{SK}_{V_j,V_i} \mid\mid T_1 \mid\mid T_2)$. After the calculation of these parameters, V_j sends the message $\mathrm{Msg}_2 = \{M_3, M_4, \mathrm{CT}^*_{V_j}, T_2\}$ to V_i through public channel.

ACVV4: Upon the arrival of Msg_2 from V_j at time T_2^*, V_i first verifies the timeliness of T_2 by using the condition: $|T_2 - T_2^*| \leq \Delta T$, and if it matches, V_i computes $\mathrm{CT}_{V_j} = \mathrm{CT}^*_{V_j} \oplus h(r_{V_i} \cdot Q_{V_j} \mid\mid T_1)$, $A_{V_j} = M_3 \oplus h(s_{V_i} . Q_{V_j} \mid\mid \mathrm{CT}_{V_j} \mid\mid T_1)$ to verify the certificate of V_j as $\mathrm{CT}_{V_j} . G = Q_{\mathrm{TA}} + h(Q_{\mathrm{TA}} \mid\mid Q_{V_j}) . N_{V_j}$. If it holds, the received certificate is the original one. V_i again computes the session key shared with V_j as $\mathrm{SK}_{V_i,V_j} = h(A_{V_i} \mid\mid A_{V_j} \mid\mid \mathrm{CT}_{V_i} \mid\mid \mathrm{CT}_{V_j} \mid\mid T_1 \mid\mid T_2)$. Then, V_i computes $M_4' = h(\mathrm{SK}_{V_i,V_j} \mid\mid T_1 \mid\mid T_2)$ and checks if $M_4' = M_4$. If it is valid, V_j is authenticated with V_i and the computed session key SK_{V_i,V_j} is correct. Next, V_i proceeds for the generation of a fresh timestamp value T_3 to estimate the session key verifier as $MV_{V_i,V_j} = h(\mathrm{SK}_{V_i,V_j} \mid\mid T_3)$ for sending the message $\mathrm{Msg}_3 = \{MV_{V_i,V_j}, T_3\}$ to V_j through public channel.

ACVV5: After receiving Msg_3 from V_i at time T_3^*, V_j first verifies the timeliness of T_3 as the condition: $|T_3 - T_3^*| \leq \Delta T$. If it holds, V_j computes the session key verifier as $MV_{V_j,V_i} = h(\mathrm{SK}_{V_j,V_i} \mid\mid T_3)$ and checks if $MV_{V_j,V_i} = MV_{V_i,V_j}$. If the values are same, V_j infers that the estimated session key by V_i is the genuine one. At the end of this phase, both V_i and V_j establish the same session key SK_{V_i,V_j} $(= \mathrm{SK}_{V_j,V_i})$ for their secure communication. Various exchanged messages during the access control and key management phase are also summarized in Fig. 5.

Access control phase between vehicles	
Smart vehicle (V_i) $\{RID_{V_i}, TC_{V_i}, s_{V_i}, CT_{V_i}\}$	Smart vehicle (V_j) $\{RID_{V_j}, TC_{V_j}, s_{V_j}, CT_{V_j}\}$
Select a random nonce r_{V_i} and timestamp T_1. Compute $A_{V_i} = h(RID_{V_i} \mid\mid TC_{V_i} \mid\mid r_{V_i} \mid\mid T_1)$, $R_{V_i} = r_{V_i} \cdot G$, $CT^*_{V_i} = CT_{V_i} \oplus h(r_{V_i} \cdot Q_{V_j} \mid\mid T_1)$, $M_1 = A_{V_i} \oplus h(s_{V_i} \cdot Q_{V_j} \mid\mid R_{V_i} \mid\mid T_1)$, $M_2 = s_{V_i} + h(M_1 \mid\mid CT^*_{V_i} \mid\mid Q_{TA} \mid\mid Q_{V_i}) * r_{V_i}$ $\pmod q$. $Msg_1 = \{M_1, M_2, R_{V_i}, CT^*_{V_i}, T_1\}$ $\longrightarrow$	
	Checks if $\|T_1 - T_1^*\| \leq \Delta T$? If so, then verify $M_2 \cdot G = Q_{V_i} +$ $h(M_1 \mid\mid CT^*_{V_i} \mid\mid Q_{TA} \mid\mid Q_{V_i}) \cdot R_{V_i}$. If validated, generate random nonce r_{V_j} and current timestamp T_2, and compute $CT_{V_i} = CT^*_{V_i} \oplus h(r_{V_j} \cdot Q_{V_i} \mid\mid T_1)$. Verify if $CT_{V_i} \cdot G = Q_{TA} + h(Q_{TA} \mid\mid Q_{V_i}) \cdot N_{V_i}$. If so, compute $A_{V_j} = h(RID_{V_j} \mid\mid TC_{V_j} \mid\mid r_{V_j} \mid\mid T_2)$, $M_3 = A_{V_j} \oplus h(s_{V_j} \cdot Q_{V_i} \mid\mid CT_{V_j} \mid\mid T_1)$. Derive $A_{V_i} = M_1 \oplus h(s_{V_j} \cdot Q_{V_i} \mid\mid R_{V_i} \mid\mid T_1)$, $CT^*_{V_j} = CT_{V_j} \oplus h(r_{V_j} \cdot Q_{V_i} \mid\mid T_1)$, and session key SK_{V_j,V_i} $= h(A_{V_i} \mid\mid A_{V_j} \mid\mid CT_{V_i} \mid\mid CT_{V_j} \mid\mid T_1 \mid\mid T_2)$, and session key verifier $M_4 = h(SK_{V_j,V_i} \mid\mid T_1 \mid\mid T_2)$. $Msg_2 = \{M_3, M_4, CT^*_{V_j}, T_2\}$ $\longleftarrow$
Checks if $\|T_2 - T_2^*\| \leq \Delta T$? If yes, then derive $CT_{V_j} = CT^*_{V_j} \oplus h(r_{V_i} \cdot Q_{V_j} \mid\mid T_1)$, $A_{V_j} = M_3 \oplus h(s_{V_i} \cdot Q_{V_j} \mid\mid CT_{V_j} \mid\mid T_1)$, and verify $CT_{V_j} \cdot G = Q_{TA} + h(Q_{TA} \mid\mid Q_{V_j}) \cdot N_{V_j}$. If verified, then compute $SK_{V_i,V_j} = h(A_{V_i} \mid\mid A_{V_j} \mid\mid CT_{V_i} \mid\mid CT_{V_j} \mid\mid T_1 \mid\mid T_2)$, and check if $h(SK_{V_j,V_i} \mid\mid T_1 \mid\mid T_2) = M_4$? If all successfully verified, pick timestamp T_3 and compute session key verifier as $MV_{V_i,V_j} = h(SK_{V_i,V_j} \mid\mid T_3)$. $Msg_3 = \{MV_{V_i,V_j}, T_3\}$ $\longrightarrow$	
	Check if $\|T_3 - T_3^*\| \leq \Delta T$? If yes, then compute $MV_{V_j,V_i} = h(SK_{V_j,V_i} \mid\mid T_3)$. Verify if $MV_{V_i,V_j} = MV_{V_j,V_i}$? If yes, then store the session key.
Both V_i and V_j establish the same session key SK_{V_i,V_j} $(= SK_{V_j,V_i})$.	

Fig. 5. Synopsis of V2V access control and key establishment.

2) Access Control Between Vehicles V_j and RSU_l: In this phase, we discuss the access control procedure between a cluster head V_j and a roadside unit RSU_l to share the real-time roadside information received from other vehicles in the network or sensed by itself. The entire process executes as follows.

VRP1: V_j proceeds for the generation of a random secret $rs_1 \in Z_q^*$ and a fresh timestamp value t_1 to compute $X_1 = h(\mathrm{RID}_{V_j} \mid\mid \mathrm{TC}_{V_j} \mid\mid r_{s_1} \mid\mid s_{V_j} \mid\mid t_1)$, $X_1 = h(\mathrm{RID}_{V_j} \mid\mid \mathrm{TC}_{V_j} \mid\mid rs_1 \mid\mid s_{V_j} \mid\mid t_1)$, $X_2 = X_1 \cdot G$, $X_3 = h(X_2 \mid\mid \mathrm{CT}_{V_j} \mid\mid t_1)$, and $\mathrm{CT}^*_{V_j} = \mathrm{CT}_{V_j} \oplus h(s_{V_j} \cdot Q_{\mathrm{RSU}_l} \mid\mid X_2 \mid\mid t_1)$. V_j sends the message $\mathrm{MSG}_1 = \{X_2, X_3, \mathrm{CT}^*_{V_j}, t_1\}$ to RSU_l via public channel.

VRP2: Upon the arrival of MSG_1 at time t_1^*, RSU_l first verifies the timeliness of t_1 through equation: $|t_1 - t_1^*| \leq \Delta t$, where the "maximum transmission delay" is given by Δt. If it holds, RSU_l drives $\mathrm{CT}_{V_j} = \mathrm{CT}^*_{V_j} \oplus h(s_{\mathrm{RSU}_l} \cdot Q_{V_j} \mid\mid X_2 \mid\mid t_1)$ and verifies if $h(X_2 \mid\mid \mathrm{CT}_{V_j} \mid\mid t_1) = X_3$. If it is valid, RSU_l verifies $\mathrm{CT}_{V_j} \cdot G = Q_{\mathrm{TA}} + h(Q_{\mathrm{TA}} \mid\mid Q_{V_j}) \cdot N_{V_i}$. If this verification happens successfully, it selects a random nonce rs_2 and timestamp t_2 to compute $X_4 = h(\mathrm{RID}_{\mathrm{RSU}_l} \mid\mid \mathrm{TC}_{\mathrm{RSU}_l} \mid\mid rs_2 \mid\mid s_{\mathrm{RSU}_l} \mid\mid t_2)$, $X_5 = X_4 \cdot G$. RSU_l further computes the session key as $\mathrm{SK}_{RV} = h(X_4 \cdot X_2 \mid\mid t_1 \mid\mid t_2)$ and other important parameters like $\mathrm{CT}^*_{\mathrm{RSU}_l} = \mathrm{CT}_{\mathrm{RSU}_l} \oplus h(s_{\mathrm{RSU}_l} \cdot Q_{V_j} \mid\mid X_2 \mid\mid t_2)$, $X_6 = h(X_5 \mid\mid \mathrm{SK}_{RV} \mid\mid \mathrm{CT}_{\mathrm{RSU}_l} \mid\mid t_2)$. After these calculations, RSU_l sends the message $\mathrm{MSG}_2 = \{X_5, X_6, \mathrm{CT}^*_{\mathrm{RSU}_l}, t_2\}$ to V_j via public channel.

VRP3: Upon the arrival of MSG_2 at time t_2^*, V_j first verifies timeliness of t_2 with the help the condition: $|t_2 - t_2^*| \leq \Delta t$. If it holds, V_j computes $\mathrm{CT}_{\mathrm{RSU}_l} = \mathrm{CT}^*_{\mathrm{RSU}_l} \oplus h(s_{V_j} \cdot Q_{\mathrm{RSU}_l} \mid\mid X_2 \mid\mid t_2)$ and verifies the certificate of RSU_l as $\mathrm{CT}_{\mathrm{RSU}_l} \cdot G = Q_{\mathrm{TA}} + h(Q_{\mathrm{TA}} \mid\mid Q_{\mathrm{RSU}_l}) \cdot N_{\mathrm{RSU}_l}$. If it holds, V_j computes the session key $\mathrm{SK}_{\mathrm{VR}} = h(X_1 \cdot X_5 \mid\mid t_1 \mid\mid t_2)$ and verifies the session key by $h(X_5 \mid\mid \mathrm{SK}_{\mathrm{VR}} \mid\mid \mathrm{CT}_{\mathrm{RSU}_l} \mid\mid t_2) = X_6$. If it happens successfully, V_j selects a new timestamp t_3, and the session key verifier as $X_7 = h(\mathrm{SK}_{\mathrm{VR}} \mid\mid t_3)$. Next, V_j sends the message $\mathrm{MSG}_3 = \{X_7, t_3\}$ to RSU_l via public channel.

Access control phase between vehicle and RSU	
Smart vehicle (V_j) $\{RID_{V_j}, TC_{V_j}, s_{V_j}, CT_{V_j}\}$	Road-side unit (RSU_l) $\{RID_{RSU_l}, TC_{RSU_l}, s_{RSU_l}, CT_{RSU_l})\}$
Select a random nonce rs_1 and timestamp t_1. Compute $X_1 = h(RID_{V_j} \|\|TC_{V_j} \|\|rs_1 \|\|s_{V_j} \|\|t_1)$, $X_2 = X_1 \cdot G$, $X_3 = h(X_2 \|\|CT_{V_j} \|\|t_1)$, $CT^*_{V_j} = CT_{V_j} \oplus h(s_{V_j} \cdot Q_{RSU_l}\|\| X_2\|\| t_1)$. $MSG_1 = \{X_2, X_3, CT^*_{V_j}, t_1\}$ $\xrightarrow{\hspace{4cm}}$	
	Verify if $\|t^*_1 - t_1\| < \Delta t$? If so, then derive $CT_{V_j} = CT^*_{V_j} \oplus h(s_{RSU_l} \cdot Q_{V_j}\|\| X_2\|\| t_1)$. Check if $h(X_2 \|\|CT_{V_j} \|\|t_1) = X_3$? If valid, verify $CT_{V_j} \cdot G = Q_{TA} + h(Q_{TA} \|\|Q_{V_j}) \cdot N_{V_i}$. If verified, then select random nonce rs_2 and timestamp t_2. Compute $X_4 = h(RID_{RSU_l}$ $\|\|TC_{RSU_l} \|\|rs_2 \|\|s_{RSU_l} \|\|t_2)$, $X_5 = X_4 \cdot G$, session key $SK_{RV} = h(X_4 \cdot X_2 \|\|t_1 \|\|t_2)$, $CT^*_{RSU_l} = CT_{RSU_l} \oplus h(s_{RSU_l} \cdot Q_{V_j}\|\| X_2\|\| t_2)$, $X_6 = h(X_5 \|\|SK_{RV} \|\|CT_{RSU_l} \|\|t_2)$. $MSG_2 = \{X_5, X_6, CT^*_{RSU_l}, t_2\}$ $\xleftarrow{\hspace{4cm}}$
Check if $\|t^*_2 - t_2\| < \Delta t$? If so, then derive $CT_{RSU_l} = CT^*_{RSU_l} \oplus h(s_{V_j} \cdot Q_{RSU_l}\|\| X_2\|\| t_2)$. Verify if $CT_{RSU_l} \cdot G = Q_{TA} + h(Q_{TA} \|\|Q_{RSU_l}) \cdot N_{RSU_l}$? Compute session key $SK_{VR} = h(X_1 \cdot X_5 \|\|t_1 \|\|t_2)$, and verify if $h(X_5 \|\|SK_{VR} \|\|CT_{RSU_l} \|\|t_2) = X_6$? If so, pick new timestamp t_3 and compute session key verifier $X_7 = h(SK_{VR} \|\|t_3)$. $MSG_3 = \{X_7, t_3\}$ $\xrightarrow{\hspace{2cm}}$	
	Check if $\|t^*_3 - t_3\| < \Delta t$? If so, verify $h(SK_{RV} \|\|t_3) = X_7$? If so, accept session key.
Both V_j and RSU_l establish the same session key $SK_{RV}(= SK_{VR})$	

Fig. 6. Summary of V2RSU access control and key establishment.

VRP4: Upon the arrival of MSG_3 at time t^*_3, RSU_l verifies timeliness of t_3 by $|t_3 - t^*_3| \leq \Delta t$. If it holds, RSU_l computes and verifies if $h(\text{SK}_{RV} \mid\mid t_3) = X_7$. The successful verification of this condition enforces RSU_l to conclude that V_j has calculated the session key correctly. If it is satisfied, both the entities will store the calculated session key $\text{SK}_{RV}(= \text{SK}_{\text{VR}})$ for their secure communication. Various exchanged messages during the access control and key management phase are also summarized in Fig. 6.

Remark 2: It is essential to mention that RSU_l and CS_k can use their "ECC-based private–public keys pairs" for their secure communication. This is because the entities, such as RSU_l and CS_k, are resource-rich devices deployed in ITS.

D. Dynamic Vehicle Addition Phase

Addition of a new vehicle to network, say V_i^{new} happens using the following steps.

DVA1: TA generates a private key of the new smart vehicle V_i^{new} as $s_{V_i}^{\text{new}} \in Z_q^*$ and computes its corresponding public key as $Q_{V_i}^{\text{new}} = s_{V_i}^{\text{new}}.G$. TA then selects $\text{ID}_{V_i}^{\text{new}}$ as the identity of V_i^{new} and calculates the corresponding pseudoidentity of V_i^{new} as $\text{RID}_{V_i}^{\text{new}} = h(\text{ID}_{V_i}^{\text{new}}||s_{\text{TA}})$ and the temporal credential of V_i^{new} as $\text{TC}_{V_i}^{\text{new}} = h(\text{ID}_{V_i}^{\text{new}} \mid\mid \text{RTS}_{V_i}^{\text{new}} \mid\mid s_{V_i}^{\text{new}} \mid\mid s_{\text{TA}}||\text{RID}_{\text{TA}})$, where $\text{RTS}_{V_i}^{\text{new}}$ is the registration timestamp of V_i^{new}.

DVA2: TA proceeds for the generation of a random temporary identity of V_i^{new} as $TID_{V_i}^{\text{new}}$ and a random secret $n_{V_i}^{\text{new}} \in Z_q^*$ to compute its corresponding public parameter as $N_{V_i}^{\text{new}} = n_{V_i}^{\text{new}}.G$. Now, the TA calculates the certificate of V_i^{new} as $\text{CT}_{V_i}^{\text{new}} = s_{\text{TA}} + h(h(\text{RID}_{V_i}^{\text{new}} \mid\mid \text{TC}_{V_i}^{\text{new}}) \mid\mid Q_{\text{TA}} \mid\mid Q_{V_i}^{\text{new}}) * n_{V_i}^{\text{new}} \pmod q$. It is noted that $n_{V_i}^{\text{new}} \in Z_q^*$ is different for distinct vehicles. The TA also announces $N_{V_i}^{\text{new}}$ as public.

DVA3: TA stores $\{\text{RID}_{V_i}^{\text{new}}, \text{TC}_{V_i}^{\text{new}}, (s_{V_i}^{\text{new}}, Q_{V_i}^{\text{new}}), \text{CT}_{V_i}^{\text{new}}, E_q(u, v), G, h(\cdot)\}$ in the memory of onboard unit $\text{OBU}_{V_i}^{\text{new}}$ of V_i^{new} before its deployment. TA deletes the sensitive values, like $n_{V_i}^{\text{new}}$ and $\text{RTS}_{V_i}^{\text{new}}$ from its database, and makes the public parameters publicly available. TA also sends the registration information $\{\text{RID}_{V_i}^{\text{new}}\}$ of V_i^{new} to RSU_l securely via the pre-shared secret key K_{TA,RSU_l}.

Timestamp	Block generation time
Last hash	Hash (using SHA-256) value of previous block
Hash	Hash value of current block
Data	N_t number of transactions
Proposer	The creator of the block
Merkle_hash	Merkle tree root of all transactions
Signature	Signature on the block
Sequence No.	Sequence number of the block
Prepare message	A finite number of messages in prepared pool
Commit message	A finite number of messages in commit pool

Fig. 7. Structure of a block to be added into blockchain (adapted from [39] and [40]).

E. Block Creation, Verification, and Addition Phase

In this phase, we elaborate block creation, addition, and verification phase for the proposed scheme. An *RSU* securely sends the data in the form of transactions to the CS network, where the CSs form a Peer-to-Peer (P2P) CS network. Once the transaction is broadcasted to the network, it can be loaded into the transactions pool which is maintained by each peer node in the network. When the transactions pool reaches to a predefined transactions threshold value, a leader is elected by a round-robin fashion from the network, and constructs a block as shown in Fig. 7 and executes a voting-based consensus mechanism using the "practical Byzantine fault tolerance (PBFT)" consensus algorithm [38]. After performing the PBFT, the proposed block will be added to the blockchain.

The details description of a block addition with execution of the voting-based PBFT algorithm is as follows.

1) Once a proposer is elected through the process of round-robin, the proposer broadcasts the generated block to the entire CS network.
2) The follower receives the block and verifies the previous block hash, current block hash, data (also known as transactions) with their own transactions pool, Merkle_hash (Merkle hash of all the transactions in the block), and signature on the block.
3) If all the verifications go successfully, the followers send the validation message to each other and also to the proposer, which is stored into the prepared message pool.
4) Every follower receives the validation message from others and checks their own prepared message pool maintained by themselves.
5) Once the prepared message pool reaches to a predefined threshold value for commitment purpose, the proposer sends a commit message to other followers.
6) Other followers receive the messages and maintain their own commit message pools, and if the pool reaches to the predefined threshold value for block addition, they can add the proposed block into their own local ledgers.

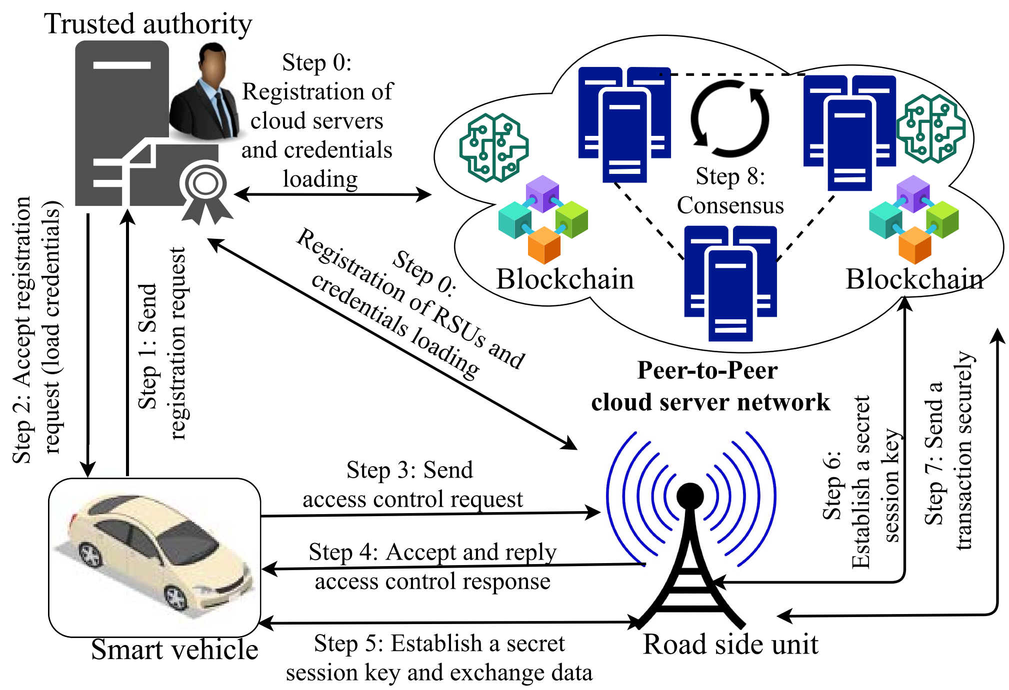


Fig. 8. Overall process diagram of the proposed framework.

After that, they broadcast the committed messages to the network.

7) Finally, the block is added and the process will again start for new block mining.

The overall process diagram of the proposed framework is given in Fig. 8. It provides a snapshot of all the above-mentioned phases, like registration, access control and key establishment, and blockchain creation. Step 0 is related to the registration of RSUs and CSs. Steps 1 and 2 are related to the registration of smart vehicles. After the successful registration of these entities, the respective credentials are loaded in their memory. Steps 3–5 are used for the access control and key establishment process of a vehicle and RSU. Similar steps are used for the access control and key establishment process of a vehicle with its neighbor vehicles. Step 6 is used for the key establishment between RSU and CS. RSU sends the transactions securely to CS using step 7. Consensus and blockchain implementation is finally performed using step 8.

Remark 3: The reason behind the use of the PBFT consensus mechanism, which is mostly used in the consortium blockchains over other public blockchain-based Proof-of-Work (PoW) and Proof-of-Stake (PoS) consensus algorithms is that PBFT is much efficient as compared to PoW and PoS in terms of computation and energy. Since the PBFT can be also used for consortium blockchains, we have chosen the voting-based PBFT algorithm which is explained in Section V-E.

Remark 4: Since the blockchain is a resource-consuming technology, it is not good to execute blockchain-related tasks at the end devices (i.e., smart vehicles). Instead of that, we use RSUs, which are resource-rich devices having high communication, computation, and storage capabilities for the creation of partial blocks, and then the associated miner node (i.e., CS) will create the full block from the received partial block. The CSs are also resource rich devices. Thus, the blockchain mining-related tasks are performed at the P2P CSs network by the CSs. After the successful execution of all steps in the proposed scheme, the blockchain is implemented at the P2P CS network. As a result, this will not have any adverse effect on the performance of the smart vehicles. Therefore, the proposed scheme does not have any effects on the performance and working of the smart vehicles.

Remark 5: It is worth noticing that the public blockchain has been incorporated in the proposed security framework of smart transportation. The blockchain technology makes such a designed framework more secure, reliable, and decentralize. We claim that the smart transportation security is fortified through the public blockchain in the framework due to the following reason. In order to update any transactions inside a block into the blockchain, an adversary needs to update or modify the following contents: 1) "last hash" which is the previous block hash; 2) "Merkle tree root" which contains the hash of all the transactions put in the block; and 3) the elliptic curve digital signature on the block. Since the signature is created by the block creator's private key, it is computationally infeasible to change the signature without having the private key of the signer. All these checks will confirm the verifier that the block is genuine and no transactions are modified by the adversary. As a result, though the transactions (information) are public in the blocks, they cannot be updated, deleted, or modified by the adversary. Hence, the smart transportation security is provided through the blockchain technology.

VI. Security Analysis of the Proposed Framework

We assess the robustness of the proposed PBSCF-ITS against the following attacks.

A. Replay Attack

For the access control and key management procedures, PBSCF-ITS uses three-type messages. All these messages are computed along with freshly generated timestamps and random secrets (nonces), which are also verified upon their arrival at the receiver's side. If an adversary $\mathcal{A}$ tries to replay the old messages, the malicious event can be easily detected by the receiving node by checking timestamps as ΔT is typically a small value. Hence, PBSCF-ITS prevents the replay attack against the passive adversary $\mathcal{A}$.

B. Man-in-the-Middle and Impersonations Attacks

Let an adversary $\mathcal{A}$ intercepts the messages Msg_1, Msg_2, and Msg_3, MSG_1, MSG_2, and MSG_3 from the public channels to launch MiTM attack. To perform this task, $\mathcal{A}$ may generate a random secret $r^a_{V_i} \in Z^*_q$ and a current timestamp T^a_1 and computes $A^a_{V_i} = h(\text{RID}_{V_i} \,||\, \text{TC}_{V_i} \,||\, r^a_{V_i} \,||\, T^a_1)$, $R^a_{V_i} = r^a_{V_i} \cdot G$, $\text{CT}^*_{V_i} = \text{CT}_{V_i} \oplus h(r^a_{V_i} \cdot Q_{V_j} \,||\, T^a_1)$, $M^a_1 = A^a_{V_i} \oplus h(s_{V_i} \cdot Q_{V_j} \,||\, R^a_{V_i} \,||\, T^a_1)$, $M^a_2 = s_{V_i} + h(M^a_1 \,||\, \text{CT}^*_{V_i} \,||\, Q_{\text{TA}} \,||\, Q_{V_i}) * r^a_{V_i} \pmod q$, where $Q_{\text{TA}} = s_{\text{TA}}.G$, $Q_{V_i} = s_{V_i}.G$, $\text{RID}_{V_i} = h(\text{ID}_{V_i}||s_{\text{TA}})$, $\text{RID}_{\text{TA}} = h(\text{ID}_{\text{TA}}||s_{\text{TA}})$, $\text{TC}_{V_i} = h(\text{ID}_{V_i} \,||\, \text{RTS}_{V_i} \,||\, s_{V_i} \,||\, s_{\text{TA}} \,||\, \text{RID}_{\text{TA}})$, RTS_{V_i} is the registration timestamp of V_i. However, $\mathcal{A}$ is not able to compute various components present in the messages Msg_1, Msg_2, and Msg_3 as they are based on secrets s_{TA}, s_{V_i}, s_{V_j}, n_{V_i}, n_{V_j} and pseudoidentities RID_{V_i} and RID_{TA}. To determine s_{TA}, s_{V_i}, and n_{V_i} from Q_{TA}, Q_{V_i}, and N_{V_i}, respectively, $\mathcal{A}$ needs to solve the computationally hard elliptic curve discrete logarithm problem (ECDLP) which is not possible for $\mathcal{A}$ in polynomial time. Thus, $\mathcal{A}$ cannot modify Msg_1 or other remaining messages. In this way, in PBSCF-ITS, $\mathcal{A}$ will not be able to launch the MiTM attack. Similarly, one can also prove that PBSCF-ITS prevents the MiTM attacks during communications between V_i and RSU_l. On the other hand, $\mathcal{A}$ cannot launch impersonation attacks on the proposed PBSCF-ITS on behalf of the legitimate entities, such as V_i, V_j, and RSU_l because the secret credentials possessed by V_i, V_j, and RSU_l cannot be obtained by the adversary $\mathcal{A}$.

C. Anonymity Preservation

In PBSCF-ITS, the secret credentials such as keys and real or pseudoidentities are not exchanged in the plaintext format. Thus, $\mathcal{A}$ does not have a chance to abuse the anonymity of the exchanged messages. Moreover, each message contains the fresh timestamp and distinct random secret numbers. Hence, PBSCF-ITS preserves anonymity property.

D. Ephemeral Secret Leakage and Privileged-Insider Attacks

The significance of the "ephemeral secret leakage (ESL) attack under the CK-adversary model" is that it tells whether a designed security scheme protects the session key or not. If the session key is computed with the help of long-term secrets as well as short-term secrets, it has potential to defend "ESL attack under the CK-adversary model." In the CK-adversary model, an adversary $\mathcal{A}$ has potential to steal the session states and session secret values. In the proposed PBSCF-ITS, the computed session keys (SK_{V_i,V_j} and SK_{RV}) use both long term secrets (identities and secret keys) along with short-term secrets (random nonces) of different parties. However, these secret values are not known to $\mathcal{A}$. In the absence of the permanent (long-term) secrets, it is infeasible for $\mathcal{A}$ to calculate the session key with having only short-term secrets through session hijacking attacks.

A privileged-insider user of the TA cannot compute the session key because most of the sensitive information are deleted from the TA's database after successful registration of registered entities. Moreover, the session keys are distinct for each session. This implies that even if a session key in a specific session is compromised, the future and previous established session keys are secure. Thus, PBSCF-ITS is resilient against ESL attack and privileged-insider attack along with preservation of both forward and backward secrecy properties.

E. Stolen Verifier Attack

In the proposed scheme, registration information is stored in the secure database (memory) of CS_k. Furthermore, we do not store any of the sensitive information in their memory directly. For example, RSU_l stores information $\{\text{RID}_{\text{RSU}_l}$, TC_{RSU_l}, $(s_{\text{RSU}_l}$, $Q_{\text{RSU}_l})$, CT_{RSU_l}, $E_q(u, v)$, G, $h(\cdot)\}$ in its memory. CS_k stores information RID_{CS_k}, TC_{CS_k}, $(s_{\text{CS}_k}, Q_{\text{CS}_k})$ in the secured region of its database. The similar mechanism is also used in the other secure cryptosystem like the RSA or ECC based systems to thwart the attempts of stolen verifier attack. Therefore, in proposed scheme required data is not available to $\mathcal{A}$ to launch the other associated attacks, i.e., the sensitive credentials guessing, impersonation, and unauthorized session key computation. Hence, the proposed scheme is able to prevent the stolen verifier attack.

F. Vehicle Physical Capture Attack

In PBSCF-ITS, OBU of a vehicle stores information $\{\text{RID}_{V_i}$, TC_{V_i}, $(s_{V_i}$, $Q_{V_i})$, $\text{CT}_{V_i}\}$ in its memory. $\mathcal{A}$ can steal an OBU physically to extract sensitive information from its memory through power analysis attack [32]. However, the information is distinct in every OBU. But, the credentials which are stored in other noncompromised OBUs are unique and distinct and it will not be of much help to the adversary. The extracted information from compromised OBU will not be further helpful in deriving of the session keys among other noncompromised smart vehicles as well as between the smart vehicles and their CSs. Hence, PBSCF-ITS is resilient against vehicle physical capture attack.

VII. Practical Blockchain Implementation

The real-time blockchain simulation has been executed over a system configuration which is considered as a CS setting with the environment setting: "Ubuntu 18.04.3 LTS, Intel Core i5-8400 CPU @ 2.80 GHz× 6, Memory 7.6 GiB, OS type

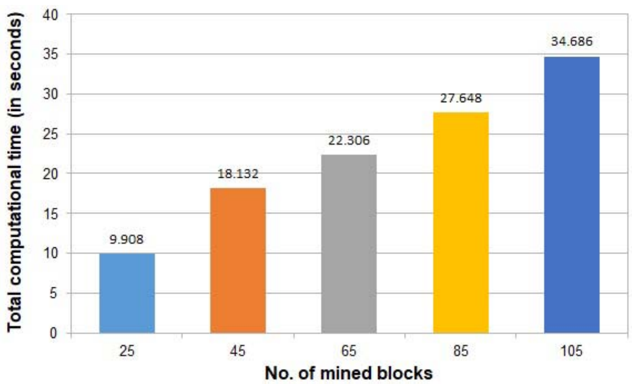


Fig. 9. Simulation results on computational time versus number of mined blocks.

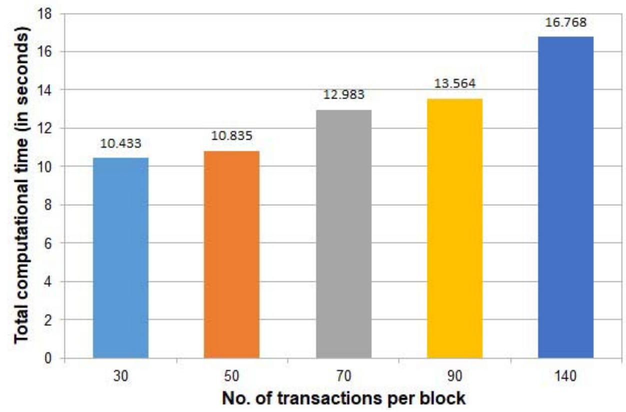


Fig. 10. Simulation results on computational time versus number of transactions per block.

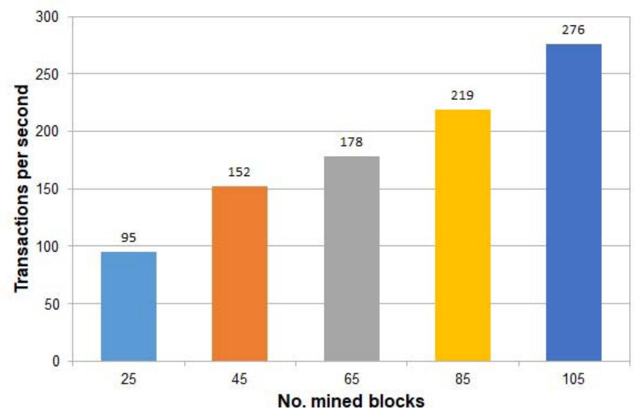


Fig. 11. Simulation results on TPS versus number of mined blocks.

64-bit, disk size 152.6 GB." The script was written in "node.js with VS CODE 2019."

Since the blockchain technology is a distributed system, the simulation is executed over a virtually created distributed servers platform. In the system, we considered the number of distributed servers (known as distributed peer nodes) as 11, and these servers create a distributed Peer-to-Peer (P2P) CS network. The peer nodes can communicate or share the information by a message passing manner, where each of the CSs has a consistent local ledger. Each ledger has the same type of data, which is similar to each other. In this simulation time, we utilized the node.js technology for creating the distributed servers as well as the messages passing process. Here, the messages indicate the created blocks which will be added into the blockchain. In addition, for the block mining process (block verification and addition into the blockchain), we implemented the voting-based "PBFT consensus algorithm" for the distributed technology. In the blockchain technology, the blocks can be added into the blockchain and each block contains a finite number of transactions. The blockchain holds a chain of various number of blocks. We examined three cases: 1) the first case contains the varied number of blocks (where each block holds a finite number of transactions) which will be added into a blockchain and we measured the time (called the total computational time in seconds) for mining the blocks using the voting-based "PBFT consensus algorithm"; 2) the second case has a varied number of transactions which are loaded into a block, and a finite number of those blocks is added into a fixed-size blockchain; and 3) the third case having a varied number of mined blocks and the transactions processed per second (TPS) was then calculated. The simulations were executed under the following three scenarios.

1) *Case 1:* In this case, we considered a fixed number of transactions for each block in the blockchain as 47. We then varied only the blockchain size, which means the number of blocks is varied. The simulation outcomes reported in Fig. 9 show the "total computational time (in seconds) versus the number of blocks mined into the blockchain." The values of computational time are 9.908, 18.132, 22.306, 27.648, and 34.686 s, for 25, 45, 65, 85 and 105 blocks to be mined, respectively. The results clearly show that whenever the size of the chain (blockchain) increases, the computational time also increases. It is worth noticing that the computational time values increase linearly with the increasing number of mined blocks.
2) *Case 2:* In this case, we considered "a fixed number of mined blocks in each blockchain as 33." The outcomes reported in Fig. 10 indicate that "the total computational time (in seconds) versus the number of transactions loaded in a block." In this case, the values of computational time are 10.433, 10.835, 12.983, 13.564, and 16.768 s for 30, 50, 70, 90 and 140 transactions containing in a block, respectively. Similar to case 1, the results signify that the computational time values increase linearly when the number of transactions per block increases.
3) *Case 3:* In this case, we estimated the values of TPS for the various number of mined blocks. The simulation outcomes reported in Fig. 11 show the values of TPS are 95, 152, 178, 219, and 276, for 25, 45, 65, 85, and 105 mined blocks, respectively. It is also observed that the values of TPS increase with the increasing number of mined blocks. This happens due to the addition of more number of blocks into the blockchain.

TABLE I
COMMUNICATION COST COMPARISON WITH RELATED PRIOR WORKS

Scheme	No. of messages	Total cost (in bits)
Liu *et al.* (2018) [27]	3	2752
Jiang *et al.* (2020) [41] (V2I initial authentication)	5	4992
Jiang *et al.* (2020) [41] (V2I handover authentication)	3	1888
Moghadam *et al.* (2020) [42]	4	3648
Ali *et al.* (2020) [43]	3	3424
Ever (2020) [44]	6	5344
Farooq*et al.* (2020) [45]	6	4032
PBSCF-ITS: Case 1	3	2208
PBSCF-ITS: Case 2	3	2016

TABLE II
AVERAGE EXECUTION TIME (IN MILLISECONDS) FOR CRYPTOGRAPHIC PRIMITIVES USING MIRACL

Primitive	Scenario 1: Raspberry PI (in milliseconds)	Scenario 2: Server (in milliseconds)
T_h	0.309	0.055
T_{mtp}	0.385	0.114
T_{senc}	0.018	0.003
T_{sdec}	0.014	0.003
T_{ecm}	2.288	0.674
T_{eca}	0.016	0.002
T_{bp}	32.084	4.716
T_{mul}	0.011	0.002
T_{add}	0.010	0.001
T_{exp}	0.228	0.039

TABLE III
COMPUTATION COST COMPARISON WITH RELATED PRIOR WORKS

Scheme	Smart device (OBU/CH/Vehicle)	Server (RSU/CS/TA/KGC)
Liu *et al.* [27]	$7T_{ecm} + 2T_{eca}$ $+6T_h + 3T_{mul} \approx 17.935$ ms	$4T_{ecm} + 3T_{eca} + 4T_h$ $+2T_{mul} + T_{bp} \approx 7.642$ ms
Jiang *et al.* [41] (V2I initial authentication)	$8T_{ecm} + 4T_{mtp} + 6T_{bp}$ $+4T_{senc}/T_{sdec} \approx 212.412$ ms	$6T_{ecm} + 2T_{mtp} + 3T_{bp}$ $+4T_{senc}/T_{sdec} \approx 18.432$ ms
Jiang *et al.* [41] (V2I handover authentication)	$5T_{ecm} + 2T_{mtp} + 3T_{bp}+$ $2T_{senc}/T_{sdec} + 2T_{mul} + T_{add}$ ≈ 108.526 ms	$5T_{ecm} + 3T_{mtp} + 3T_{bp}$ $+2T_{senc}/T_{sdec}$ ≈ 17.866 ms
Moghadam *et al.* [42]	$5T_h + 4T_{ecm} + 2T_{senc}/T_{sdec}$ ≈ 10.729 ms	$5T_h + 2T_{ecm} + 2T_{senc}/T_{sdec}$ ≈ 1.629 ms
Ali *et al.* [43]	$18T_h + T_{fe} + T_{senc}$ ≈ 7.868 ms	$7T_h + 3T_{senc}/T_{sdec}$ ≈ 0.394 ms
Ever [44]	$9T_h + 2T_{bp}+$ $2T_{mtp} + 3T_{ecm}$ ≈ 74.583 ms	$6T_h + 3T_{bp}+$ $2T_{mtp} + 3T_{ecm}$ ≈ 16.728 ms
Farooq *et al.* [45]	$T_h + 2T_{bp} + 3T_{ecm} + T_{mul}$ ≈ 71.352 ms	$T_h + 2T_{bp} + 6T_{ecm} + T_{mul}$ $+2T_{mtp} \approx 13.761$ ms
PBSCF-ITS	$8T_h + 5T_{ecm}$ ≈ 13.912 ms	$7T_h + 5T_{ecm}$ ≈ 3.755 ms

VIII. COMPARATIVE STUDY

In this section, we provide the details of conducted comparison among the proposed scheme and other similar existing schemes. The proposed scheme is compared with the other related schemes, such as Liu *et al.* [27], Jiang *et al.* [41], Moghadam *et al.* [42], Ali *et al.* [43], Ever [44], and Farooq *et al.* [45]. The details of comparisons are provided as follows.

A. Communication Costs Comparison

For the comparison of communication costs, we consider the sizes of different cryptographic operation as follows. We consider 256, 160, 160, and 320 bits for cryptographic one way hash function, random nonce/secret value, various identities, and ECC point multiplication, respectively. The communication costs of Liu *et al.* [27], Jiang *et al.* [41] (V2I initial authentication), Jiang *et al.* [41] (V2I handover authentication), Moghadam *et al.* [42], Ali *et al.* [43], Ever [44], and Farooq *et al.* [45] are estimated as 2752, 4992, 1888, 3648, 3424, 5344, and 4032 bits, respectively. Moreover, the communication costs for the proposed scheme are 2208 bits (for case 1: V2V) and 2016 bits (for case 2: V2RSU), respectively. From Table I, it is clear that the proposed scheme requires less communication costs as compared to the other existing schemes.

B. Computation Costs Comparison

For the estimation of computation costs, we use the average execution time (in milliseconds) values of cryptographic primitives, which were computed through "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL)" [46]. Let T_h, T_{mtp}, T_{senc}/T_{sdec}, T_{ecm}/T_{eca}, T_{bp}, $T_{\text{mul}}/T_{\text{add}}$, and T_{exp} signify the execution time required for one-way hash function, map-to-point, symmetric encryption/decryption, bilinear pairing, modular multiplication/addition, and modular exponentiation, respectively.

The execution time of various cryptographic operations are provided in Table II. In Table II, Scenario-1 is taken for resource constrained devices, i.e., sensing devices, IoT sensors, etc., under the setting: "Raspberry PI 3 B+ Rev 1.3, Ubuntu 20.04 LTS, 64-bit OS, 1.4-GHz Quad-core processor, cores 4, 1-GB RAM." On the other side, Scenario-2 is taken for resource rich devices, i.e., servers, gateway nodes, etc., under the setting: "Ubuntu 18.04.4 LTS, with 7.7-GiB memory, Intel Core processor-8565U, CPU @ 1.80 GHz $\times$ 8, 64-bit OS type and disk size 966.1 GB." We executed each cryptographic operation for 100 times and measured the minimum, maximum, and average execution time in milliseconds.

The values of computation time for the proposed scheme are 14.839 and 21.085 ms in case-1 (for V2V communication) and 13.912 ms 3.755 in case-2 (for V2RSU communication). From Table III, it is clear that the proposed scheme requires less computation cost as compared to some other schemes. Though the computation cost of the proposed scheme is higher than some of the schemes, but it can be accepted as it provides "more security and extra functionality features."

C. Comparison of Security and Functionality Features

The security and functionality features of the proposed scheme and other schemes, such as Liu *et al.* [27], Jiang *et al.* [41] (V2I initial authentication), Jiang *et al.* [41] (V2I handover authentication), Moghadam *et al.* [42],

TABLE IV
FUNCTIONALITY AND SECURITY ATTRIBUTES DIFFERENTIATION

Attribute	[27]	[41]	[42]	[43]	[44]	[45]	PBSCF-ITS
F_1	✓	✓	✓	✓	✓	×	✓
F_2	✓	✓	✓	✓	✓	✓	✓
F_3	✓	✓	✓	✓	✓	✓	✓
F_4	✓	✓	✓	✓	✓	✓	✓
F_5	✓	✓	✓	✓	✓	✓	✓
F_6	✓	✓	✓	✓	✓	✓	✓
F_7	✓	×	✓	×	×	×	✓
F_8	✓	✓	✓	✓	✓	✓	✓
F_9	×	×	×	×	×	✓	✓
F_{10}	×	×	×	×	×	×	✓
F_{11}	×	×	×	✓	×	×	✓
F_{12}	×	×	×	×	×	×	✓
F_{13}	×	×	×	✓	×	×	✓
F_{14}	✓	✓	✓	✓	✓	×	✓

F_1: "replay attack"; F_2: "man-in-the-middle attack"; F_3: "mutual authentication"; SF_4: "key agreement"; F_5: "device/vehicle impersonation attack"; F_6: "RSU/server impersonation attack"; F_7: "anonymity"; F_8: "resilience against device (vehicle) physical capture attack"; F_9: "ESL attack under the CK-adversary model"; F_{10}: "formal security verification using AVISPA tool"; F_{11}: "support dynamic node (vehicle/RSU) addition phase"; F_{12}: "support blockchain-based solution"; F_{13}: "support formal security analysis under ROR model"; F_{14}: "privileged-insider attack"
✓: "a scheme is secure or it supports an attribute"; ×: "a scheme is insecure or it does not support an attribute".

Ali *et al.* [43], Ever [44], and Farooq *et al.* [45] are compared in Table IV. From Table IV, it is clear that other existing schemes are vulnerable to various potential attacks and lack in functionality features. However, the proposed scheme provides desired level of security and also supports extra functionality features. Therefore, the proposed scheme seems better than the other existing schemes.

IX. CONCLUSION AND FUTURE DIRECTIONS

In this article, we aimed to design an effective access control and key management solution for Big data analytics-endowed ITS, called PBSCF-ITS. The security analysis of PBSCF-ITS proves its resilience against various types of potential attacks. A rigorous comparative study with existing related schemes reveals that PBSCF-ITS can provide more security and functionality features than the existing counterparts. Therefore, PBSCF-ITS can be a suitable mechanism for deployment in a secure communication for Big data analytics-endowed ITS.

In the future, we try to include more functionality features in the proposed framework. Moreover, we also aim to include the testbed experiments of the proposed framework in a real-time environment to measure its performance with the actual settings.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers and the Associate Editor for their valuable feedback on this article, which helped them to improve its quality and presentation.

REFERENCES

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.
[2] O. Pribyl, P. Pribyl, M. Lom, and M. Svitek, "Modeling of smart cities based on ITS architecture," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 4, pp. 28–36, Nov. 2019.
[3] L. F. Herrera-Quintero, J. C. Vega-Alfonso, K. B. A. Banse, and E. C. Zambrano, "Smart ITS sensor for the transportation planning based on IoT approaches using serverless and microservices architecture," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 17–27, Apr. 2018.
[4] A. Mehrabi and K. Kim, "Low-complexity charging/discharging scheduling for electric vehicles at home and common lots for smart households prosumers," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 348–355, Aug. 2018.
[5] M. Milosevic, M. Z. Bjelica, T. Maruna, and N. Teslic, "Software platform for heterogeneous in-vehicle environments," *IEEE Trans. Consum. Electron.*, vol. 64, no. 2, pp. 213–221, May 2018.
[6] J. Dey, W. Taylor, and S. Pasricha, "VESPA: A framework for optimizing heterogeneous sensor placement and orientation for autonomous vehicles," *IEEE Consum. Electron. Mag.*, vol. 10, no. 2, pp. 16–26, Mar. 2021.
[7] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1064–1078, Feb. 2019.
[8] S. Kaffash, A. T. Nguyen, and J. Zhu, "Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis," *Int. J. Prod. Econ.*, vol. 231, Jan. 2021, Art. no. 107868.
[9] Y. Lian, G. Zhang, J. Lee, and H. Huang, "Review on big data applications in safety research of intelligent transportation systems and connected/automated vehicles," *Accid. Anal. Prevent.*, vol. 146, Oct. 2020, Art. no. 105711.
[10] Y. Wang, J. Yu, B. Yan, G. Wang, and Z. Shan, "BSV-PAGS: Blockchain-based special vehicles priority access guarantee scheme," *Comput. Commun.*, vol. 161, pp. 28–40, Sep. 2020.
[11] F. Li *et al.*, "Blockchain-based trust management in distributed Internet of Things," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–12, Dec. 2020.
[12] S. Jangirala, A. K. Das, and A. V. Vasilakos, "Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 7081–7093, Nov. 2020.
[13] I. A. Umoren, S. S. A. Jaffary, M. Z. Shakir, K. Katzis, and H. Ahmadi, "Blockchain-based energy trading in electric-vehicle-enabled microgrids," *IEEE Consum. Electron. Mag.*, vol. 9, no. 6, pp. 66–71, Nov. 2020.
[14] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
[15] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. Park, "AKM-IoV: Authenticated key management protocol in fog computing-based Internet of Vehicles deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8804–8817, Oct. 2019.
[16] G. S. Aujla and A. Jindal, "A decoupled blockchain approach for edge-envisioned IoT-based healthcare monitoring," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 491–499, Feb. 2021.
[17] R. Chaudhary, A. Jindal, G. S. Aujla, N. Kumar, A. K. Das, and N. Saxena, "LSCSH: Lattice-based secure cryptosystem for smart healthcare in smart cities environment," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 24–32, Apr. 2018.
[18] R. Chaudhary, A. Jindal, G. S. Aujla, S. Aggarwal, N. Kumar, and K.-K. R. Choo, "BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system," *Comput. Security*, vol. 85, pp. 288–299, Aug. 2019.
[19] Y. Li, Y. Yu, C. Lou, N. Guizani, and L. Wang, "Decentralized public key infrastructures atop blockchain," *IEEE Netw.*, vol. 34, no. 6, pp. 133–139, Nov./Dec. 2020.
[20] A. Vangala, B. Bera, S. Saha, A. K. Das, N. Kumar, and Y. H. Park, "Blockchain-enabled certificate-based authentication for vehicle accident detection and notification in intelligent transportation systems," *IEEE Sensors J.*, vol. 21, no. 14, pp. 15824–15838, Jul. 2021, doi: 10.1109/JSEN.2020.3009382.
[21] Y. Liu, Y. Wang, and G. Chang, "Efficient privacy-preserving dual authentication and key agreement scheme for secure V2V communications in an IoV paradigm," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2740–2749, Oct. 2017.
[22] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, "An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1621–1632, May 2019.

[23] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.

[24] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based Industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[25] B. S. Egala, A. K. Pradhan, V. R. Badarla, and S. P. Mohanty, "Fortified-chain: A blockchain based framework for security and privacy assured Internet of Medical Things with effective access control," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11717–11731, Jul. 2021.

[26] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "PoBT: A lightweight consensus algorithm for scalable IoT business blockchain," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2343–2355, Mar. 2020.

[27] J. Liu, Q. Li, R. Sun, X. Du, and M. Guizani, "An efficient anonymous authentication scheme for Internet of Vehicles," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, 2018, pp. 1–6.

[28] S. R. Pokhrel, Y. Qu, S. Nepal, and S. Singh, "Privacy-aware autonomous valet parking: Towards experience driven approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5352–5363, Aug. 2021.

[29] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Federated learning for data privacy preservation in vehicular cyber-physical systems," *IEEE Netw.*, vol. 34, no. 3, pp. 50–56, May/Jun. 2020.

[30] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology (EUROCRYPT)*, B. Pfitzmann, Ed. Innsbruck, Austria: Springer, 2001, pp. 453–474.

[31] A. K. Das, "A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks," *Int. J. Inf. Security*, vol. 11, no. 3, pp. 189–211, 2012.

[32] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[33] A. K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, and X. Huang, "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Security Commun. Netw.*, vol. 9, no. 16, pp. 3670–3687, 2016.

[34] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Gener. Comput. Syst.*, vol. 91, pp. 475–492, Feb. 2019.

[35] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.

[36] S. Liu, J. Yu, Y. Xiao, Z. Wan, S. Wang, and B. Yan, "BC-SABE: Blockchain-aided searchable attribute-based encryption for cloud-IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7851–7867, Sep. 2020.

[37] W. E. May, "Secure hash standard," U.S. Dept. Commerce, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. FIPS PUB 180-1, 2015. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf (accessed Jan. 2020).

[38] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002.

[39] B. Bera, A. K. Das, and A. K. Sutrala, "Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in Internet of Drones environment," *Comput. Commun.*, vol. 166, pp. 91–109, Jan. 2021.

[40] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, "Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled Internet of Drones environment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9097–9111, Aug. 2020.

[41] Y. Jiang, S. Ge, and X. Shen, "AAAS: An anonymous authentication scheme based on group signature in VANETs," *IEEE Access*, vol. 8, pp. 98986–98998, 2020.

[42] M. F. Moghadam, M. Nikooghadam, M. A. B. Al Jabban, M. Alishahi, L. Mortazavi, and A. Mohajerzadeh, "An efficient authentication and key agreement scheme based on ECDH for wireless sensor network," *IEEE Access*, vol. 8, pp. 73182–73192, 2020.

[43] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43711–43724, 2020.

[44] Y. K. Ever, "A secure authentication scheme framework for mobile-sinks used in the Internet of Drones applications," *Comput. Commun.*, vol. 155, pp. 143–149, Apr. 2020.

[45] S. M. Farooq, S. M. S. Hussain, T. S. Ustun, and A. Iqbal, "Using ID-based authentication and key agreement mechanism for securing communication in advanced metering infrastructure," *IEEE Access*, vol. 8, pp. 210503–210512, 2020.

[46] "MIRACL Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library." [Online]. Available: https://github.com/miracl/MIRACL (accessed Oct. 2020).

Mohammad Wazid (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the International Institute of Information Technology Hyderabad, Hyderabad, India, in 2017.

He is currently working as an Associate Professor with the Department of Computer Science and Engineering, and the Head of Cyber security and IoT Research Group, Graphic Era University, Dehradun, India. He has published over 100 research papers in international journals and conferences. His current research interests include security, remote user authentication, Internet of Things, and blockchain.

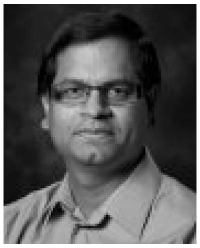
Basudeb Bera received the M.Tech. degree in computer science and data processing from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 2017. He is currently pursuing the Ph.D. degree in computer science and engineering with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad, India.

He has published over 25 research papers in international journals and conferences. His research interests are cryptography, network security, and blockchain technology.

Ashok Kumar Das (Senior Member, IEEE) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 1996, 2000, and 2008, respectively.

He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad, India. His Google Scholar H-index is 61 and i10-index is 184 with over 11 300 citations. His research interests include cryptography, system and network security, blockchain, security in Internet of Things, Internet of Vehicles, Internet of Drones, smart grids, smart city, cloud/fog computing, intrusion detection, and AI/ML security. He has authored over 285 papers in international journals and conferences in the above areas, including over 245 reputed journal papers.

Dr. Das was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of IEEE SYSTEMS JOURNAL, *Journal of Network and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), *Journal of Cloud Computing* (Springer), *Cyber Security and Applications* (Elsevier), *IET Communications*, *KSII Transactions on Internet and Information Systems*, and *International Journal of Internet Technology and Secured Transactions* (Inderscience), and has served as a program committee member in many international conferences.



Saraju P. Mohanty (Senior Member, IEEE) received the bachelor's degree (Hons.) in electrical engineering from the Orissa University of Agriculture and Technology, Bhubaneswar, India, in 1995, the master's degree in systems science and automation from the Indian Institute of Science Bangalore, Bengaluru, India, in 1999, and the Ph.D. degree in computer science and engineering from the University of South Florida, Tampa, FL, USA, in 2003.

He is a Professor with the University of North Texas, Denton, TX, USA. His research is in "Smart Electronic Systems" which has been funded by National Science Foundation, Semiconductor Research Corporation, U.S. Air Force, IUSSTF, and Mission Innovation. He has authored 400 research articles, four books, and invented seven granted/pending patents. His Google Scholar H-index is 44 and i10-index is 181 with 8645 citations.

He is a recipient of 13 best paper awards, the Fulbright Specialist Award in 2020, the IEEE Consumer Electronics Society Outstanding Service Award in 2020, the IEEE-CS-TCVLSI Distinguished Leadership Award in 2018, and the PROSE Award for Best Textbook in Physical Sciences and Mathematics category in 2016. He has delivered 11 keynotes and served on 12 panels at various international conferences. He was the Editor-in-Chief of the *IEEE Consumer Electronics Magazine* from 2016 to 2021 and serves on the editorial board of six journals/transactions.

Minho Jo (Senior Member, IEEE) received the B.A. degree from the Department of Industrial Engineering, Chosun University, Gwangju, South Korea, in 1984, and the Ph.D. degree from the Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA, in 1994.

He is the Professor with the Department of Computer Convergence Software, Korea University, Sejong, South Korea, where he is the Director of the IoT & AI Lab. His current research interests include IoT, blockchain, artificial intelligence and deep learning, big data, network security, cloud/edge computing, wireless energy harvesting, and autonomous vehicles.

Prof. Jo is a recipient of the 2018 IET Best Paper Premium Award by the United Kingdom's Royal Institute of Engineering and Technology. He is one of the founders of the Samsung Electronics LCD Division. He is the Founder and the Editor-in-Chief of *KSII Transactions on Internet and Information Systems* (SCI/JCR and SCOPUS indexed). He currently serves for the South Korea's Presidential Commission on Policy Planning. He is currently an Associate Editor of IEEE SYSTEMS JOURNAL, IEEE ACCESS, and IEEE INTERNET OF THINGS JOURNAL. He is an Editor of IEEE WIRELESS COMMUNICATIONS.