Blockchain-Based Efficient Access Control With Handover Policy in IoV-Enabled Intelligent Transportation System

Sandip Roy, Sourav Nandi, Raj Maheshwari, Sachin Shetty, *Senior Member, IEEE*, Ashok Kumar Das, *Senior Member, IEEE*, and Pascal Lorenz, *Senior Member, IEEE*

***Abstract*—Recent advances in Internet technology and IoT devices have facilitated researchers to foster a wide range of Intelligent Transportation Systems (ITS) that improve the quality of automated transportation by addressing real-time safety and traffic management issues. The participating ITS agents, such as smart cars and roadside equipment, are required to communicate urgently through an open (unsecured) wireless channel in an unattended setting. To address the security issues, several vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) authentication and access control protocols have been proposed in recent times. However, fast-moving vehicles need to set up frequent authentication with different roadside units, which induces high computation and communication overheads. Consequently, it becomes a bottleneck for the resource-limited vehicle onboard unit devices. As the blockchain supports decentralized storage with data integrity and transparency, in this article, we design a secure and lightweight Internet of Vehicles (IoV)-enabled blockchain-based access control protocol with a handover authentication facility (we call it BACHP-IoV, in short). The handover authentication mechanism exploits no computation-costly cryptographic primitives. Once the transactions or messages have been securely gathered by a roadside unit (RSU), RSU_j, residing in a group of vehicles Vh_i, will form a partial block, which is later forwarded to a cloud server node in the Peer-to-Peer (P2P) cloud servers blockchain network for converting it into a full block. Next, the full blocks are mined using a voting-based consensus algorithm. In addition, the in-charge trusted authority $\mathcal{TA}$ uploads information about the registered vehicles, such as randomized masked passwords and random secrets, to the blockchain. Thus, an RSU_j can check the authenticity of a particular vehicle as well. We prove the security strength of the proposed BACHP-IoV by using the well-known Real-or-Random (ROR)-based random oracle model, the ProVerif 2.03 simulation tool, and informal security analysis. We have implemented the proposed BACHP-IoV through network simulator 3 (NS-3) and blockchain, and the simulation results demonstrate that BACHP-IoV is practical in a real-life scenario. A detailed comparative analysis also shows that BACHP-IoV provides significantly better security and efficiency than the existing competing schemes.**

***Index Terms*—Intelligent Transportation System (ITS), Internet of Vehicles (IoV), blockchain, access control, authentication handover, NS3 simulation, security.**

Manuscript received 14 February 2023; revised 6 April 2023 and 12 August 2023; accepted 1 October 2023. Date of publication 13 October 2023; date of current version 14 March 2024. This work was supported in part by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) under Contract W911NF-20-2-0277 and in part by the U.S. Army Research Laboratory, National Science Foundation under Grants 2219742 and 2131001. The review of this article was coordinated by Prof. Xianbin Cao. *(Corresponding authors: Ashok Kumar Das; Pascal Lorenz.)*

Sandip Roy is with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA 23435 USA (e-mail: sroy@odu.edu).

Sourav Nandi is with the Department of Computer Science and Engineering, National Institute of Technology Durgapur, Durgapur 713209, India (e-mail: sn.21p10119@mtech.nitdgp.ac.in).

Raj Maheshwari is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India (e-mail: raj.maheshwari@students.iiit.ac.in).

Sachin Shetty is with the Virginia Modeling, Analysis and Simulation Center, Department of Computational Modeling and Simulation Engineering, Old Dominion University, Suffolk, VA 23435 USA (e-mail: sshetty@odu.edu).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India, and also with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA 23435 USA (e-mail: ashok.das@iiit.ac.in).

Pascal Lorenz is with the University of Haute Alsace, 68008 Colmar, France (e-mail: lorenz@ieee.org).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVT.2023.3322637, provided by the authors.

Digital Object Identifier 10.1109/TVT.2023.3322637

I. Introduction

OVER the last few years, rapid technological development and research have been carried out in the fields of the Internet of Things (IoT), smart grids, smart city-based applications, etc. Communication among various things, such as smart devices, vehicles, and so on, is an inherent part of Information and Communications Technology (ICT)-based infrastructures. These ICT-based infrastructures can play a useful role in e-health care, automatic transportation, and various other fields. Intelligent Transportation System (ITS) is one such application area that is growing in popularity. ITS facilitates the sensing, analyzing, and controlling of a safe and reliable driving experience by protecting responsible entities and messages from various cyber attackers [1], [2], [3].

Vehicular ad hoc networks (VANETs) are a key part of the intelligent transportation systems (ITS) framework. VANETs are created by applying the principles of mobile ad-hoc networks (MANETs) to the domain of vehicular networks. The VANETs system can be represented as a two-layer architecture. The upper layer is made up of trusted authority ($\mathcal{TA}$) and the lower layer is made up of roadside units (RSU) and vehicles. Each vehicle is equipped with an onboard unit (OBU). Typically, an ITS consists of various RSUs. An RSU is in charge of validating the signals sent by the moving vehicles that make up its group or cluster, which is established dynamically. The

OBU is sometimes regarded as a transportable unit. An OBU can function whether a car or a person is moving or stationary.

VANETs consist of various entities, where the vehicles are connected in an ad-hoc manner for exchanging data with each other. Generally, VANET uses different technologies to ensure a non-identical way of communication like vehicle to vehicle (V2V), vehicle to road-side unit (V2RSU), and road-side unit to the cloud server (RSU2C) [4], [5]. On the other side, the Internet of Vehicles (IoV) spans a larger network that involves various entities, like "humans", "smart things" and other "heterogeneous networks". Thus, IoV provides real-time information exchange on the roads in a smart city among the "deployed vehicles and sensors", "vehicles and vehicles", "vehicles and roads", and "vehicles and personal devices" with the help of various wireless communication technologies [6].

Blockchain technology facilitates an immutable, decentralized, secure, and anonymous way of storing data, providing robustness against various types of information security-related attacks [7], [8]. As a result, in recent years, the use of blockchain for smart vehicles that enable ITS communication has become a natural choice [9]. In a real-life scenario, due to the continuously increasing number of vehicles, the traffic load might exceed the capacity of the existing infrastructure. Such a system authorizes the formation of secure data management and sharing techniques for data at rest or in motion. In general, VANET uses various non-identical ways of communication like vehicle to vehicle (V2V), vehicle to a roadside unit (V2RSU), and roadside unit to the cloud server (RSU2C) [4], [5].

The use of blockchain technology has become more common in recent years across a range of applications. Singh and Jindal devised the blockchain-based strategy of the independently edge-envisioned ecosystem [10]. Here, adjacent edge devices are employed to generate the decoupling blocks on the blockchain. This could offer a safe cryptosystem for smart healthcare in a smart city [11]. Zhou et al. [12] introduced a blockchain-based volunteer edge cloud for IoT applications to solve the growing issues of large-scale IoT applications. Their suggested approach increases the adaptability of software development while reducing the complexity of IoT devices.

Energy trading calculation, which is done using real-time processing, is one of the key requirements of the tactile Internet. To overcome such challenges, Choo et al. [13] proposed an energy trading scheme for electric vehicles (EVs) using blockchain, which can perform securely. This scheme also ensures resilience against a single point of failure. Inherently, IoV-enabled ITS requires frequent message communications among various entities of the system through insecure wireless channels. This allures potential adversary agents to launch several active and passive security attacks like eavesdropping, revealing, modifying, deleting, or intercepting communicated messages [14], [15]. In a real-time scenario of an ITS, the presence of such security vulnerabilities may lead to several catastrophic results. Thus, the guaranteed enforcement of detection and prevention of such security attacks is an indispensable requirement.

Inherently, IoV-enabled ITS requires frequent message communications among various entities of the system through insecure wireless channels. This allures potential adversary agents to launch several active and passive security attacks like eavesdropping, revealing, modifying, deleting, or intercepting communicated messages [14], [15]. In a real-time scenario of an ITS, the presence of such security vulnerabilities may lead to catastrophic results like fatal accidents, etc. Thus, the guaranteed enforcement of detection and prevention of such security attacks is an indispensable requirement.

Very recently, the blockchain has been used as a cutting-edge technology to address a set of intriguing issues in several IoV-based applications. Vishwakarma et al. [16] used the "modified practical byzantine fault tolerance (mPBFT) consensus algorithm" to design a blockchain-based security protocol for secure communication and storage in a software-defined network (SDN)-enabled IoV. Their experimental results show that the storage, computation, and communication overheads are noticeably low as compared to those for the other state-of-the-art schemes. Lin et al. [17] proposed a new parking-sharing network using the blockchain to build a decentralized, but trustworthy network. Their proposed scheme exploits the "theory of planned behavior (TPB)" to find an effective way to get timely information about available parking lots. For VANETs, Zhou et al. [18] proposed an efficient certificateless conditional privacy-preserving authentication (CPPA) scheme. They indicated and proved the security loopholes of Ali et al.'s protocol [19]. The authors experimentally demonstrated the efficiency of their scheme over the existing solutions, in terms of verification, signing, and bandwidth requirements.

In ITS, migration and authentication with other RSUs are continuous processes. Without the presence of handover authentication, the whole authentication process needs to be executed from the beginning. As a result, the computation cost is high and the network efficiency is low. Frequent repetition of the authentication processes leads to a huge burden for the vehicles and thus fails to assure real-time communication [20]. As a consequence, designing a lightweight and secure access control with an inter-RSU authentication handover protocol still remains a challenging problem in research.

The OBU's biometric sensing unit's recent advancements have encouraged the development of various biometric-assisted enhanced authentication schemes in VANET. While both face and fingerprint biometrics are used in practice [21], fingerprint biometrics is gaining traction in VANET, particularly for driver identification and driving characteristic representation [22], [23]. In this article, we propose a biometric-assisted, blockchain-based access control and authentication handover protocol for an IoV environment. If the biometric templates are used in plaintext, it is seemingly feasible to get them compromised [24]. As a result, the proposed scheme contemplates the use of a stable and dependable biometric fuzzy extractor technique to protect the user's biometric information [25], [26]. Note that the users do not need to imprint biometrics during inter-RSU authentication handover.

A. Motivation and Problem Statement

In addition to decentralized nature, the blockchain technology has benefits in the realm of data exchange, such as the provision of data provenance, and data integrity. During the

authentication handover process, the inter-RSU information exchange and storage should be done in the utmost secured way. If the handover information passed from one RSU to another is erased or forged, it may compromise network security. To facilitate communication and verification during the authentication-handover process, RSUs can function as nodes in a consortium blockchain. Whenever a moving vehicle enters into the range of a new RSU, a transaction composed of authentication-handover information is added to the blockchain. Furthermore, because each transaction on the blockchain bears the RSU's signature and is protected by the blockchain's inherent properties, it cannot be altered or falsified in any way. As a result, other RSUs can verify the authenticity of the vehicle by using a transaction's information stored in the blockchain. This makes us think that the blockchain technology could be a way to make V2I handover authentication safe and lightweight [27].

In recent years, a lot of researches have been conducted by applying blockchain technology in VANETs, such as vehicle key management [28], trustworthiness [29], and batch verification [30]. However, very little research focus has been put into the inter-RSU authentication handover problem. Recently, Son et al. [27] proposed a blockchain-based V2I handover authentication scheme. However, we found that their scheme is vulnerable to de-synchronization attacks. Furthermore, this method has a significant computational overhead during the initial key-establishment phase, and it does not support password and biometric update mechanisms, and also does not provide a way for the dynamic addition of vehicles into the system. Hence, we find that the design of a secure, robust, and lightweight blockchain-enabled inter-RSU authentication handover scheme is still a challenging research problem in ITS.

The blockchain is decentralized in nature and possesses the ability for data sharing, data transparency, and integrity. These features motivate us to use the blockchain consisting of the cloud server nodes, where the RSUs provide a way to access and share information for vehicles during inter-RSU handover authentication, apart from the real-traffic information and other private and confidential information as transactions. As a result, a consortium blockchain has been used for data storage. Once a vehicle shares a session key with its attached RSU, it can maneuver the information from the blockchain for handover authentication with other RSUs as well. This will provide us with the use of active blockchain during the access control and authentication process as in the case of other IoT networking environments [31]. An active blockchain helps in storing as well as retrieving the data and secret credentials at the same without the need for deployment of smart contracts, even during the progress of registration and authentication processes. In addition, it has been observed by Mitra et al. [32] that there is a significant impact on Big Data analytics on the data stored in the non-blockchain storage versus the data stored in the blockchain. This is because when the data is simply stored in a non-blockchain platform, there is a possibility that the semi-trusted cloud servers may corrupt or modify the stored data using data poising attacks, whereas due to the inherent properties of the blockchain, it is difficult for an adversary to corrupt the data in the blockchain.

B. Research Contributions

The major research contributions of the proposed scheme are:

1) In an ITS environment, the proposed BACHP-IoV scheme provides an extended chaotic map and Chebyshev polynomials-based access control mechanisms between vehicles and roadside units. The comparative result indicates that the use of these extended chaotic maps makes the access control policy much more efficient than the baseline policies.
2) The BACHP-IoV scheme provides a blockchain-based seamless inter-RSU authentication handover mechanism, that does not use any computation-costly cryptosystem. The authentication handover process is based on symmetric key encryption-decryption, hash function, and XOR operations only.
3) BACHP-IoV allows an efficient way to both add and revoke vehicles from the system. Hence, the scheme is scalable and suitable for a practical real-life scenario.
4) We validate the security of the proposed BACHP-IoV scheme through a widely used random oracle model-based formal security proof. Further, using the ProVerif 1.93 simulation tool, we simulate the security strength of the proposed scheme.

C. Paper Organization

Here goes the road map for the remaining part of this article. Section II discusses on the merits and demerits of the related papers. Section III presents the network and the threat model adopted for BACHP-IoV. In Section IV, we present the proposed BACHP-IoV scheme, which contains in detail. Along with several other sub-phases, the proposed scheme contains the secure data aggregation phase and the blockchain implementation phase. The Informal security analysis and ROR oracle model-based security proof are provided in Section V. Section VI provides security verification simulation using ProVerif 2.03 tool. Various comparative studies with baseline policies are presented in Section VII. Section VIII presents the practical implementation of the proposed scheme using the NS3 simulation as well as blockchain simulation on various network attributes and scenarios. Finally, Section IX discusses the scope of future works and concludes the paper.

II. Related Work

This section illustrates the existing research paper or works related to secure authentication, key management, and blockchain in ITS. A comparison of existing state-of-art authentication schemes is given in Table I.

Pribyl et al. [2] proposed a model for smart cities based on ITS architecture but has a drawback of system complexity. In 2018, Herrera-Quintero et al. [3] designed a smart ITS sensor for the Bus Rapid Transit (BRT) systems. In this work, they build an IoT-based serverless and microservice architecture that can be used in Bus Rapid Transit (BRT) systems. Lian et al. [43] reviewed and analyzed road safeness in ITS and environment for communication of connected automated vehicles (CAV).

TABLE I
COMPARATIVE SUMMARY RECENT AUTHENTICATION AND HANDOVER SCHEMES IN INTELLIGENT TRANSPORTATION SYSTEMS

Scheme	Year	Concept applied	Major contributions	Drawbacks
Tan and Chung [33]	2020	* Bilinear pairings * Consortium blockchain for V2V group key * ECC, one-way hash function and modular exponentiation	* Certificate-less authentication for cloud-assisted VANETs * Blockchain-based group-key distribution * Dynamic group-key updating for vehicles group	* Does not support dynamic vehicle addition * Not resilient against privileged insider attack * Does not support inter-RSU authentication handover scenario
Ali et al. [19]	2021	* Elliptic curve cryptography (ECC) * One-way hash function * Certificateless cryptography (CLC)	* Certificateless short signature-based conditional privacy-preserving authentication for VANETs * Supports batch signature verification batch-verification without bilinear-pairing	* Fails to resist signature forgery attacks [18] * Cannot guarantee that the user's key pair is indeed initialized by KGC [18].
Nandy et al. [35]	2021	* Elliptic curve cryptography (ECC) * One-way hash function * Symmetric-key encryption	* Privacy-preserving key establishment in VANETs * Practical implementation using NS3 & SUMO	* Incorrect public-private key pair [34] * Suffers from clogging attack [34], [36] * Exploitation of useless pseudo-identities
Wang et al. [37]	2020	* Bilinear pairing * Consortium blockchain	* Provides blockchain assisted trustworthiness scalable computation * V2I authentication handover strategy is designed	* Bilinear pairing incurs high computation cost * Insecure to session-specific random number [38] leakage attack [38], [27] * Insecure to impersonation attack [38]
Meng et al. [39]	2021	* ECC * One-way hash function	* Support non-full key escrow authentication for V2V communication * Resist the ephemeral key leakage attack * Provide security proof in the eCK model	* Does not preserve unlinkability property [36] * Insecure to physical capture attack [36] * Does not preserve vehicle user anonymity * Does not support conditional privacy preservation
Ma et al. [40]	2020	* Bivariate polynomial * ECC-based ECDSA and ECIES algorithms * One-way hash function	* lightweight distributed storage mechanism * Blockchain based decentralized key management * Resist key tampering and collusion attacks * Automatic public key management	* No inter-RSU authentication handover * Lack of an efficient storage mechanism [41] * Suffer from scalability issue * Does not support anonymity
Feng et al. [42]	2020	* PBFT consortium blockchain * ECC and attribute-based encryption * Smart contract	* Supports traceability * Dynamic revocation of misbehaving vehicles	* Does not evaluate communication overhead * Does not evaluate storage overhead * No inter-RSU authentication handover
Son et al. [27]	2022	* PBFT consortium blockchain * ECC * AES & hash function	* Supports blockchain based V2I initial authentication and V2I handover authentication * Supports efficient vehicle revocation	* Fails against de-synchronization attack * Does not provide password change and biometric update mechanism and dynamic vehicle addition mechanism
Zhou et al. [18]	2022	* Certificateless cryptography (CLC) * ECC	* Demonstrate security drawbacks of [19] * Design certificateless conditional privacy preserving authentication for VANETs * Support conditional traceability, anonymity, and un-linkability	* No inter-RSU handover authentication * Cost of aggregate verification is high

Wazid et al. [5] designed a fog computing-based IoV deployment for communication among fog cloud servers, vehicles and RSUs, which can communicate securely. One of the pioneering research on the design of handover and pre-handover authentication protocol for VANETs was carried out by Gao et al. [44]. However, due to the exploitation of computation costly point multiplication, their scheme suffers from high-performance overhead.

Liu et al. [45] developed a dual authentication scheme for IoV which can work in different scenarios. A lightweight blockchain and trade consensus algorithm for IoT was presented by Biswas et al. [46]. The mechanism used in this algorithm allows the trades validation as well as blocks with minimal computation cost. Zhang et al. [47] proposed a model using consortium blockchain which has the property of a decentralized storage system and shared data securely. This mechanism is fully reliable for the transmitted data. In the same year, Zheng et al. proposed a scheme using blockchain for transaction storage which is secure, and it provides decentralization and transparency of the vehicles [48]. However, it does not support mutual authentication.

The design of a seamless, efficient, and lightweight handover authentication protocol is still a challenging area of research. Recently, a blockchain-endowed authentication system for VANETs was proposed by Feng et al. [42]. In their proposed model, in the absence third party, the transmitted messages can be verified by RSUs and vehicles. However, the existing research on blockchain-based VANET does not support mutual authentication. To solve this, Ma et al. [40] suggested a blockchain-assisted authentication system using distributed storage characteristic of blockchain. In their scheme, they used automatic public key revocation and updating the public key using the smart contract. However, the computation cost to generate blocks is very high and does not support the handover situation. Wang et al. [37] designed a blockchain-based reliable computation of vehicles using a bilinear pairing-based signature, and proposed a V2I handover authentication scheme, However, its computational cost is very high.

Following these seminal works, Vangala et al. [49] designed a new blockchain-enabled digital-certificate-based authentication scheme that supports both services of vehicle accident detection and notification propagation in ITS. Here, on detection of self or neighboring vehicle accidents, accident-related information is transferred by each vehicle to its nearby cluster head in a secure manner. Son et al. [27] proposed a blockchain-based authentication system that supports both initial authentication and mutual authentication. However, their scheme is vulnerable to de-synchronization attacks and incurs high computation overhead during the initial authentication phase. Moreover, this scheme does not provide a dynamic password and biometric update mechanism, and lacks a dynamic vehicle addition mechanism.

III. NETWORK AND THREAT MODELS

A. Network Model

This section briefly describes the network model that facilitates a blockchain-based handover authentication policy. The model consists of four major roles, which are trusted authority ($\mathcal{TA}$), roadside unit (RSU), vehicle, and blockchain. Fig. 1 provides the intended network model for authentication handover in the Intelligent Transportation System (ITS).

Trusted Authority ($\mathcal{TA}$): Vehicles and the RSUs should be registered before the deployment into the network with the help of a trusted authority ($\mathcal{TA}$). Note that the registration process is only one-time procedure.

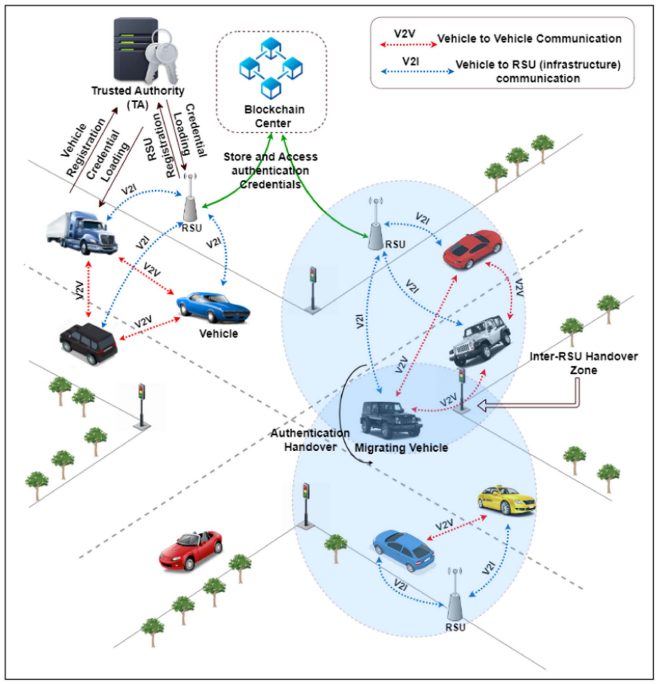


Fig. 1. Blockchain-based authentication handover model.

Road Side Unit (RSU): RSU broadcasts or exchanges traffic data to the on-board unit (OBU) which is equipped with vehicles and in its communication zone. A RSU can check to see if a vehicle is registered with the $\mathcal{TA}$. After successful authentication, RSU uploads the transactions with signed parameters into the blockchain, which can be further used to check the authentication of vehicles by RSUs. Blockchain is used to store the information as well as the messages (real-time traffic data) aggregated to RSUs from various vehicles in an ITS environment.

Vehicles: Vehicles are used as a medium for sending or receiving real-time traffic data through the attached OBU. In an ITS, dedicated short-range communications (DSRC) is an IEEE 802.11p-based technology that facilitates the direct wireless exchange of data in vehicle-to-everything (V2X) infrastructure. It is an open-source wireless communication protocol designed to facilitate high-speed wireless communication between vehicles and other infrastructures, like RSU_j [4]. As per the DSRC standard, every vehicle delivers location-based messages to its nearby vehicles and RSU, every 150-300 ms, which indicates that an RSU should verify around 650 location-based messages if there are approximately 200 vehicles within the RSU coverage range [4], [50]. Through the proposed blockchain-based authentication handover, RSU_j will be able to avoid the computational burden of redundant authentication mechanisms, thereby preventing a bottleneck. After registering to $\mathcal{TA}$, smart vehicles can execute the authentication process with RSUs or nearby vehicles through DSRC or cellular networks. As in [51], we also assume a dynamic cluster for creating different clusters of vehicles on the fly [52]. Here, each car looks for nearby vehicles that are traveling in the same direction and at the same pace. Clusters are produced by moving vehicles in the same lane that approach intersections with other lanes. According to [52], when a group of these cars has been picked, one of these vehicles is chosen to act as the cluster head.

Blockchain Center: A blockchain center is considered as a Peer-to-Peer (P2P) network of cloud server nodes (CS). Once the transactions or messages are securely gathered by an RSU_j residing in a group of vehicles Vh_i, RSU_j will form a partial block, which is later forwarded to a cloud server node in the P2P CS blockchain network for converting it into a full block. Next, the full block is mined using a voting-based consensus algorithm. Note that a public blockchain is typically used in an ITS environment. Since, in ITS, some transactions may be private and confidential, and those transactions need to be also encrypted with the public key of the block owner, say RSU_j, a consortium blockchain may be the best suited for an ITS environment. Here, we use consortium blockchain for authentication activity information, which leads to privacy preservation. $\mathcal{TA}$ uploads information about the registered vehicles, such as randomized masked passwords and random secrets to the blockchain. Thus, an RSU_j can check the authenticity of a particular vehicle as well. Due to inherent properties, a blockchain provides protection against data modification (immutability), transparency, and decentralization.

B. Threat Model

To analyze whether our proposed scheme is secured against various attacks or not, we consider the broadly used model known as the "Dolev-Yao threat (DY) model" [53] with the following:

i) Since the vehicles, RSUs, end users are not trustable, the messages can be modified, intercepted, deleted, or eavesdropped on by an adversary;
ii) An adversary can capture the OBUs of the vehicle physically to assume the sensitive data by power analysis attacks [54]. Furthermore, it can cause impersonation and man-in-the-middle attacks.

We use another *de facto* adversary model, "Canetti and Krawczyk's adversary model (CK-adversary model)" [55].

i) The CK-adversary model is similar to the DY model, but it has additional capabilities such as the compromise of secret credentials via session-hijacking attacks, secret keys, and session states.
ii) This feature makes it standard a model of exchanges that are as important as the confidentiality of particular secret confirmations from the hijacking attack session and should definitely be a small impact on the security of the suspended session key.

The $\mathcal{TA}$ is responsible for registering businesses in the network is considered a fully reliable node.

IV. The Proposed Scheme

This section presents various phases of our scheme. Table II tabulates the key notations used in the rest of the paper along with a brief description. In this article, registration of RSU phase is abbreviated as RRSU. Consequently, the ith step of this phase is denoted as RRSUi. In a similar fashion, the ith step of the vehicle registration phase is termed as VHRi, the ith step of the access control phase between a vehicle and

TABLE II
BRIEF DESCRIPTION OF KEY NOTATIONS

Symbol	Description
$\mathcal{TA}$	Trusted registration authority
Vh_i	i^{th} vehicle user
RSU_j	j^{th} road-side unit
Id_{vi}	Identity of the i^{th} vehicle user
Id_{rsu_j}	Identity of the j^{th} road-side unit
Pw_{vi}	Password chosen by Vh_i
$\mathcal{B}_{vi}$	Biometrics of user of vehicle Vh_i
$\mathcal{T}_s$	1024-bit master secret key of $\mathcal{TA}$
x_τ	1024-bit random number in $(-\infty, +\infty)$
$\mathcal{P}_\tau$	1024-bit public parameter chosen by $\mathcal{TA}$
$T_x(\cdot)$	A Chebyshev polynomial [57], [58]
MId_{vi}	masked password of Vh_i
n_{vi}	128-bit random nonce
σ_{rsu_j}	Private key of RSU_j
$T_{\sigma_{rsu_j}}(x_\tau)$	Public key of RSU_j
b_{vi}	Private key of Vh_i
$T_{b_{vi}}(x_\tau)$	Public key of Vh_i
$\mathcal{TS}_{vi}$	Current system timestamp of Vh_i
$\mathcal{TS}_{rsu_j}$	Current system timestamp of RSU_j
$\triangle T$	Maximum transmission delay of a message
$\|\|, \oplus$	Concatenation, bitwise XOR operations
$h(\cdot)$	One-way cryptographic hash function
$(\lambda_{vi}, \omega_{vi})$	Parameters generated by fuzzy extractors technique [25], [59]
$Generation(\cdot)$, $Reproduction(\cdot)$	Fuzzy extractor "probabilistic generation" functions [25]
$Sig_{RSU_j}(\cdot)$	Message signed by RSU_j using ECDSA signature

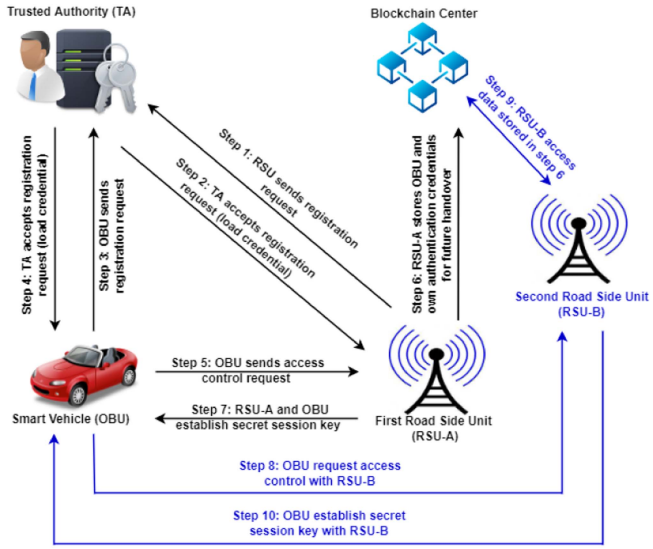


Fig. 2. Process diagram of the proposed scheme.

an RSU is referred as VRACi, the ith step of inter-RSU vehicle authentication handover is termed as AHVRi, the ith step of new vehicle addition phase is written as NVAi, and finally, the ith step of vehicle revocation phase is denoted as VhRev$i$.

A. High-Level Description

Fig. 2 shows the overall process diagram of the proposed BACHP-IoV scheme. The scheme contains the following phases, namely (1) registration of roadside units and smart vehicles, (2) access control, (3) blockchain-based vehicle authentication handover among RSUs, (4) dynamic vehicle addition, and (5) smart vehicle revocations.

Various phases of the proposed scheme are summarized below:

- *System Initialization Phase:* In this phase, $\mathcal{TA}$ initializes its own master secret key, necessary public parameters, own registration timestamp, and selects a suitable hash algorithm.
- *Registration Phase:* During this phase, all RSUs and smart vehicle users must register with the $\mathcal{TA}$ individually. For RSU *registration*, $\mathcal{TA}$ first selects ID, computes the secret key of RSU_j, and delivers it to RSU_j through a secure channel. For *vehicle registration*, Vh_i selects its own credentials like ID, password, and fingerprint-biometrics uses them to create a masked password, and delivers it to $\mathcal{TA}$ through a secure channel. $\mathcal{TA}$ selects the secret key of Vh_i, uses a random nonce to randomize the masked password, loads the necessary encrypted parameters into OBU_i memory, and finally saves the vehicle record with biometrics into the blockchain.
- *Access Control between Vehicles Vh_i and RSU_j:* In this phase, Vh_i and RSU_j mutually authenticate one another and prepare a shared session key for message communication. First, Vh_i submits all of its own credentials, and the scheme validates the user's identity using the OBU_i stored parameters. Following that, Vh_i computes the required hash parameter using its own secret key, RSU_j's public key, and the current timestamp and sends the necessary parameter set to RSU_j for verification via a public channel. RSU_j validates message transmission delay, employs its own secret key and Vh_i's public key, computes required hash parameters, and validates login message integrity. Following successful verification, RSU_j and Vh_i prepare a shared session key. Finally, RSU_j signs and uploads the record to the blockchain, which aids in the handover of inter-RSU authentication.
- *Inter-RSU Vehicle Authentication Handover using Blockchain:* In this phase, the protocol transfers authentication setup for an inter-RSU migrating vehicle. Vh_i sends an authentication handover request message to new roadside unit RSU_k, which is already authenticated to RSU_j. RSU_k retrieves blockchain record signed by RSU_j, and validates the request message integrity. Following successful verification, RSU_k authenticates Vh_i, and setups the shared session key with Vh_i. Finally, RSU_k updates the blockchain record with Vh_i.
- *Dynamic Vehicle Addition Phase:* The proposed scheme is scalable and can add new vehicles dynamically. In this phase, a new vehicle Vh_{new} delivers its new set of credentials to $\mathcal{TA}$ and registers in a fresh way.
- *Vehicle Revocation Phase:* This phase is responsible for the revocation of a malicious vehicle, initiated by a roadside unit RSU_x. RSU_x discovers the malicious vehicle's masked password and sends it to $\mathcal{TA}$ via a "secure channel". $\mathcal{TA}$ deletes the corresponding vehicle entry from the

Trusted authority ($\mathcal{TA}$)	**Road-side unit** (RSU_j)
Select identity Id_{rsu_j} for RSU_j.	
Compute $\sigma_{rsu_j} = h(Id_{rsu_j}\|\|h(x_\tau\|\|Rt_{rsu_j}))$.	Save received parameters
$\xrightarrow[\text{Secure channel}]{\{Id_{rsu_j},\ \sigma_{rsu_j}\}}$	$(Id_{rsu_j},\ \sigma_{rsu_j})$.
	Compute Chebyshev polynomial $T_{\sigma_{rsu_j}}(x_\tau)$.
Delete σ_{rsu_j} from own memory.	Declare $T_{\sigma_{rsu_j}}(x_\tau)$ as public key.

Fig. 3. Summary of registration of RSU_j to $\mathcal{TA}$.

blockchain and updates the vehicle user revocation list in the blockchain.

B. Detailed Description

1) 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 trusted authority $\mathcal{TA}$ selects a 1024-bit master secret key $\mathcal{T}_s$. Next, $\mathcal{TA}$ selects two 1024-bit random numbers x_τ (randomly in the interval $(-\infty, +\infty)$) and $\mathcal{P}_\tau$ as a public parameter. $\mathcal{TA}$ generates a 128-bit number Rt_τ, which signifies its registration timestamp. Furthermore, $\mathcal{TA}$ selects a one-way (collision-resistant) hash function $h(\cdot)$ (for instance, Secure Hash Algorithm (SHA-256) hashing algorithm [58] that produces 256-bit hash output), a non-singular elliptic curve $E_q(\alpha, \beta)$ with the curve: $y^2 = x^3 + \alpha x + \beta$ over a finite (Galois) field $GF(q)$, q being a sufficiently large prime with a base point G whose order will be as big as the selected prime q, and the "Elliptic Curve Digital Signature Algorithm (ECDSA)" [59].

2) Registration Phase: The participating entities must be registered before using the network services. Before functioning, all vehicles (Vhs) and roadside units (RSUs) must be registered to the TA. The registration of these entities is executed offline mode through a secure channel. Registration of different network entities is discussed below.

- *Registration of RSU:* Each road side unit RSU_j must register to the $\mathcal{TA}$ through the following steps:

Step RRSU1:

1) $\mathcal{TA}$ chooses a new and unique identity Id_{rsu_j} for RSU_j.
2) $\mathcal{TA}$ computes secret key of RSU_j as $\sigma_{rsu_j} = h(Id_{rsu_j}||h(x_\tau||Rt_{rsu_j}))$. Here, x_τ is the public parameter as described in Section IV-B1 and Rt_{rsu_j} is the registration timestamp of RSU_j.
3) $\mathcal{TA}$ delivers $\{Id_{rsu_j}, \sigma_{rsu_j}\}$ to RSU_j through a secure channel and deletes σ_{rsu_j} from its memory.

Step RRSU2:

1) RSU_j saves both (Id_{rsu_j} and σ_{rsu_j}).
2) Using the public parameter x_τ, RSU_j computes a Chebyshev polynomial $T_{\sigma_{rsu_j}}(x_\tau)$. Finally, RSU_j declares $T_{\sigma_{rsu_j}}(x_\tau)$ as its own public key.

The summary of the registration process of RSU_j to $\mathcal{TA}$ is shown in Fig. 3.

- *Registration of Smart Vehicles:* To achieve three-factor authentication, Vh_i imprints fingerprint biometrics $\mathcal{B}_{vi}$ along with ID and password. The fuzzy extractor takes $\mathcal{B}_{vi}$ from Vh_i and employs a probabilistic generation function to generate error-tolerant unique random strings $(\lambda_{vi}, \omega_{vi})$. During initial login, Vh_i generates the same λ_{vi} by combining stored ω_{vi} with a noisy user biometric input $\mathcal{B}'_{vi}$ that differs from the original biometric $\mathcal{B}_{vi}$ up to a threshold value [25], [26]. For details, the formal definition of a fuzzy extractor is defined in the supplementary material.

Each registered vehicle Vh_i must be registered to the $\mathcal{TA}$ by following the steps below:

Step VHR1:

1) Vehicle user Vh_i chooses its unique identity Id_{vi}, selects password Pw_{vi}, and imprints biometrics $\mathcal{B}_{vi}$ in its the on-board unit (OBU_{vi}).
2) Using fuzzy extractor $Generation(\cdot)$ function, Vh_i generates parameters $(\lambda_{vi}, \omega_{vi}) = Generation(\mathcal{B}_{vi})$ and computes the masked password $VPB_{vi} = h(Id_{vi}||Pw_{vi}||\lambda_{vi})$ and delivers $(Id_{vi}, \mathcal{B}_{vi}, VPB_{vi})$ to $\mathcal{TA}$ via a secured channel.

Step VHR2:

1) $\mathcal{TA}$ checks the uniqueness of the vehicle id Id_{vi} and confirms that it is not yet been used by any other registered vehicle. Next, for Vh_i, $\mathcal{TA}$ generates a 1024-bit unique secret key b_{vi}.
2) For Vh_i, $\mathcal{TA}$ chooses an 1024-bit unique number $r_{\tau_{vi}}$. Further, $\mathcal{TA}$ computes $RPB_{vi} = h(VPB_{vi}||h(b_{vi}))$, and a masked password $MId_{vi} = h(RPB_{vi}||r_{\tau_{vi}})$. Note that MId_{vi} is unique for each vehicle Vh_i.

Step VHR3:

1) $\mathcal{TA}$ computes $\alpha_{vi} = RPB_{vi} \oplus r_{\tau_{vi}}$, generates an 128-bit random nonce n_{vi} and computes $\beta_{vi} = MId_{vi} \oplus n_{vi}$.
2) $\mathcal{TA}$ computes the hash parameter $\gamma_{vi} = h(RPB_{vi}||MId_{vi}||n_{vi})$ and $\mathcal{X}_{vi} = VPB_{vi} \oplus b_{vi} = h(Id_{vi}||Pw_{vi}||\lambda_{vi}) \oplus b_{vi}$.
3) $\mathcal{TA}$ computes $h(MId_{vi}||\mathcal{P}_\tau)$ and uploads it to the blockchain. Also, $\mathcal{TA}$ saves record $< MId_{vi}, \mathcal{B}_{vi} >$ into blockchain for future access.
4) $\mathcal{TA}$ loads parameters $(\alpha_{vi}, \beta_{vi}, \gamma_{vi}, \mathcal{X}_{vi})$ into the on-board unit OBU_{vi} of Vh_i. Further, Using secret key b_{vi} and the public parameter x_τ, $\mathcal{TA}$ computes a Chebyshev polynomial $T_{b_{vi}}(x_\tau)$ and declares $T_{b_{vi}}(x_\tau)$ as public key of Vh_i. $\mathcal{TA}$ deletes secret key b_{vi} and biometric $\mathcal{B}_{vi}$ from its memory.

Step VHR4: Vh_i receives OBU_{vi} parameters $(\alpha_{vi}, \beta_{vi}, \gamma_{vi}, \mathcal{X}_{vi})$ from $\mathcal{TA}$. Vh_i adds parameter $\{\omega_{vi}\}$ into OBU_{vi}.

The summary of the registration process of smart vehicle Vh_i to $\mathcal{TA}$ is shown in Fig. 4.

In addition, after registration of the entities like RSUs and smart vehicles, for blocks construction, each RSU_j starts generating its own ECC-based private key by randomly selecting $pr_{RSU_j} \in E_q(\alpha, \beta)$ and then computing the corresponding public key $Pub_{RSU_j} = pr_{RSU_j} \cdot G$ as the elliptic curve point multiplication, where $pr_{RSU_j} \cdot G = G + G + \cdots + G(pr_{RSU_j}$ times). Similarly, for mining of the blocks in the blockchain, the "private-public key pairs $(pr_{CS_w} \in E_q(\alpha, \beta),\ Pub_{CS_w} = pr_{CS_w} \cdot G)$ are generated by the respective cloud server node CS_w" in the P2P CS blockchain network.

Remark 1: Note that, as RSU_js cannot be considered as a fully trusted unit, we do not store any secret vehicle credentials in the memory of RSU_j. Moreover, we consider OBU_{vi} is not

Vehicle user (Vh_i)	**Trusted authority** ($\mathcal{TA}$)
Select identity Id_{vi}, password Pw_{vi}.	Check availability of Id_{vi}.
Imprint biometrics $\mathcal{B}_{vi}$, and compute	Generate 1024-bit key b_{vi}.
$(\lambda_{vi}, \omega_{vi}) = Generation(\mathcal{B}_{vi})$,	Generate 1024-bit random number $r_{\tau_{vi}}$.
$VPB_{vi} = h(Id_{vi}\|\|Pw_{vi}\|\|\lambda_{vi})$	Compute $RPB_{vi} = h(VPB_{vi}\|\|h(b_{vi}))$.
$\xrightarrow[\text{Secure channel}]{\{Id_{vi}, \mathcal{B}_{vi}, VPB_{vi}\}}$	Compute $MId_{vi} = h(RPB_{vi}\|\|r_{\tau_{vi}})$.
	Compute $\alpha_{vi} = RPB_{vi} \oplus r_{\tau_{vi}}$.
	Generate 128-bit nonce n_{vi}.
	Compute $\beta_{vi} = MId_{vi} \oplus n_{vi}$,
	$\gamma_{vi} = h(RPB_{vi}\|\| MId_{vi}\|\| n_{vi})$,
	$\mathcal{X}_{vi} = VPB_{vi} \oplus b_{vi}$
	$= h(Id_{vi}\|\| Pw_{vi}\|\| \lambda_{vi}) \oplus b_{vi}$.
	Upload $h(MId_{vi}\|\|\mathcal{P}_\tau)$ into blockchain.
	Save $\{MId_{vi}, \mathcal{B}_{vi}\}$ into blockchain.
	Load $(\alpha_{vi}, \beta_{vi}, \gamma_{vi}, \mathcal{X}_{vi})$ into OBU_i.
	Using secret key b_{vi} compute
	Chebyshev polynomial $T_{b_{vi}}(x_\tau)$.
	Declare $T_{b_{vi}}(x_\tau)$ as Vh_i's public key.
Load $\{\omega_{vi}\}$ into OBU_i memory.	Delete b_{vi} and $\mathcal{B}_{vi}$ from memory.

Fig. 4. Registration of smart vehicle Vh_i to $\mathcal{TA}$.

hardware tamper-proof. So, we set and store parameters α_{vi}, β_{vi}, γ_{vi} and $\mathcal{X}_{vi}$ in a way, where adversary cannot launch stolen device attack.

3) Access Control Phase: This phase establishes secure and lightweight access control between a smart vehicle and its nearby roadside unit. After mutual authentication, Vh_i and RSU_j form the necessary shared secret session key for future message communication. Each vehicle Vh_i and RSU_j establish a shared session key among themselves for secure message communication using the following steps:

Step VRAC1:

1) Vehicle user Vh_i enters identity Id_{vi}, password Pw_{vi}, and imprints biometrics $\mathcal{B}'_{vi}$ in its the on-board unit (OBU_{vi}). Using $\mathcal{B}'_{vi}$, stored parameter ω_{vi}, and fuzzy extractor $Reproduction(\cdot)$ function, Vh_i retrieves $\lambda_{vi} = Reproduction(\mathcal{B}'_{vi}, \omega_{vi})$.
2) Vh_i retrieves $b'_{vi} = \mathcal{X}_{vi} \oplus h(Id_{vi}\|\| Pw_{vi}\|\| \lambda_{vi})$ and computes $RPB'_{vi} = h(h(Id_{vi}\|\| Pw_{vi}\|\| \lambda_{vi})\|\| h(b'_{vi}))$.
3) Vh_i uses stored parameter α_{vi} to compute $r'_{\tau_{vi}} = \alpha_{vi} \oplus RPB'_{vi}$. Further, using computed RPB'_{vi} and retrieved $r'_{\tau_{vi}}$, Vh_i generates $MId'_{vi} = h(RPB'_{vi}||r'_{\tau_{vi}})$. Also, Vh_i retrieves n'_{vi} as $n'_{vi} = \beta_{vi} \oplus MId'_{vi}$.
4) Finally, Vh_i uses all retrieves parameter to verify if stored $\gamma_{vi} = h(RPB'_{vi}\|\| MId'_{vi}\|\| n'_{vi})$. If the verification holds true, OBU_{vi} signals a successful login of the vehicle user Vh_i into the system. Otherwise, the login & authentication process in terminated.

Step VRAC2:

1) Vh_i uses its secret key b_{vi} and RSU_j's public parameter $T_{\sigma_{rsu_j}}(x_\tau)$ to compute $A_{v_i,rsu_j} = T_{b_{vi}}(T_{\sigma_{rsu_j}}(x_\tau))$ and its own Chebyshev polynomial $A_{v_i} = T_{b_{vi}}(x_\tau)$.
2) Vh_i computes $Y_1 = h(Id_{rsu_j}\|A_{v_i,rsu_j}\|\mathcal{TS}_{vi}) \oplus MId_{vi}$ and $Y_2 = h(A_{v_i,rsu_j}\|MId_{vi}\|\mathcal{TS}_{vi})$. Vh_i sends authentication request $Msg_1 = \{A_{v_i}, Y_1, Y_2, \mathcal{TS}_{vi}\}$ to RSU_j though a public channel.

Step VRAC3:

1) On receiving Msg_1 at time $\mathcal{TS}^*_{vi}$, RSU_j verifies if $|\mathcal{TS}^*_{vi} - \mathcal{TS}_{vi}| \leq \triangle T$? If the message transmission delay is more than the threshold value $\triangle T$, RSU_j discards Msg_1.
2) RSU_j uses own secret key σ_{rsu_j} and received A_{v_i} to compute $A_{rsu_j,v_i} = T_{\sigma_{rsu_j}}(A_{v_i}) = T_{\sigma_{rsu_j}}(T_{b_{vi}}(x_\tau)) = T_{b_{vi}}(T_{\sigma_{rsu_j}}(x_\tau)) = A_{v_i,rsu_j}$.
3) RSU_j own id Id_{rsu_j}, received timestamp $\mathcal{TS}_{vi}$ and computed A_{rsu_j,v_i} and generate $MId_{vi} = Y_1 \oplus h(Id_{rsu_j}||A_{rsu_j,v_i}||\mathcal{TS}_{vi})$.
4) Using retrieved MId_{vi} and public $\mathcal{P}_\tau$, RSU_j generates $h(MId_{vi}||\mathcal{P}_\tau)$. Moreover, it makes access to the blockchain ledger and searches the presence of the record $h(MId_{vi}||\mathcal{P}_\tau)$ is present. On successful trace, RSU_j checks if received $Y_2 = h(A_{rsu_j,v_i}||MId_{vi}||\mathcal{TS}_{vi})$. If verification fails, RSU_j terminates the authentication process.

Step VRAC4:

1) RSU_j generates secret number $\sigma^1_{rsu_j}$ and computes $B_{rsu_j} = T_{\sigma^1_{rsu_j}}(x_\tau)$ and $B_{rsu_j,v_i} = T_{\sigma^1_{rsu_j}}(T_{b_{vi}}(x_\tau))$
2) RSU_j computes $\Omega_{vij} = MId_{vi} \oplus h(Id_{rsu_j}\|\ m_{rsu_j})$, where m_{rsu_j} is an 128-bit random nonce.
3) RSU_j computes session key shared with Vh_i as $SK_{rsu_j,v_i} = h(Id_{vi}\|\ Id_{rsu_j}\|\ B_{rsu_j,v_i}\|\ \Omega_{vij}\|\ MId_{vi}\|\ \mathcal{TS}_{vi}\|\ \mathcal{TS}_{rsu_j})$, where $\mathcal{TS}_{rsu_j}$ refers to the current timestamp.
4) RSU_j computes $Y_3 = \Omega_{vij} \oplus h(MId_{vi}||B_{rsu_j,v_i})$ and hash parameter $Y_4 = h(Id_{vi}||Id_{rsu_j}||SK_{rsu_j,v_i})$. Finally, RSU_j sends $Msg_2 = \{B_{rsu_j}, Y_3, Y_4, \mathcal{TS}_{rsu_j}\}$ to Vh_i though a public channel.

Step VRAC5:

1) Vh_i receives RSU_j reply message Msg_2 and verifies if if $|\mathcal{TS}^*_{rsu_j} - \mathcal{TS}_{rsu_j}| \leq \triangle T$? If the message transmission delay is more than the threshold value $\triangle T$, Vh_i discards Msg_2.
2) Otherwise, Vh_i uses secret key $T_{b_{vi}}$ and received parameter B_{rsu_j} to compute
$$B_{v_i,rsu_j} = T_{b_{vi}}(T_{\sigma^1_{rsu_j}}(x_\tau)) = T_{\sigma^1_{rsu_j}}(T_{b_{vi}}(x_\tau)) = T_{\sigma^1_{rsu_j}}(T_b(x_\tau)) = B_{rsu_j,v_i}$$
3) Using computed B_{v_i,rsu_j} and received Y_3, Vh_i retrieves $\Omega'_{vi} = Y_3 \oplus h(MId_{vi}||B_{v_i,rsu_j})$.
4) Vh_i computes the session key shared with RSU_j as $SK_{v_i,rsu_j} = h(Id_{vi}\|\ Id_{rsu_j}\|\ B_{v_i,rsu_j}\|\ \Omega'_{vi}\|\ MId_{vi}\|\ \mathcal{TS}_{vi}\|\ \mathcal{TS}_{rsu_j})$.
5) Using session key SK_{v_i,rsu_j}, Vh_i computes and verifies if $Y_4 = h(Id_{vi}||Id_{rsu_j}||SK_{v_i,rsu_j})$? If this verification holds, then Vh_i generates fresh timestamp $\mathcal{TS}^1_{vi}$ and sends $Msg_3 = \{Y_5, \mathcal{TS}^1_{vi}\}$ to RSU_j though a public channel.
6) On receiving Msg_3, RSU_j checks validity of timestamp $\mathcal{TS}^1_{vi}$ and verifies if $Y_5 = h(SK_{rsu_j,v_i}||\mathcal{TS}^1_{vi})$. On successful verification, the session key is established for future secure communication. RSU_j also uploads $(\Omega_{vij}, m_{rsu_j}, Id_{rsu_j}, Sig_{RSU_j}(h(\Omega_{vij}\|\ MId_{vi}\|\ m_{rsu_j})))$ to the blockchain.

The authentication procedure between Vh_i and RSU_j is summarized in Fig. 5.

Remark 2: Vh_i uses secret key b_{vi} to generate parameters A_{v_i,rsu_j} and A_{v_i}. Note that this secret key is not stored in the OBU_{vi} device. Rather, it is computed by Vh_i in run-time way.

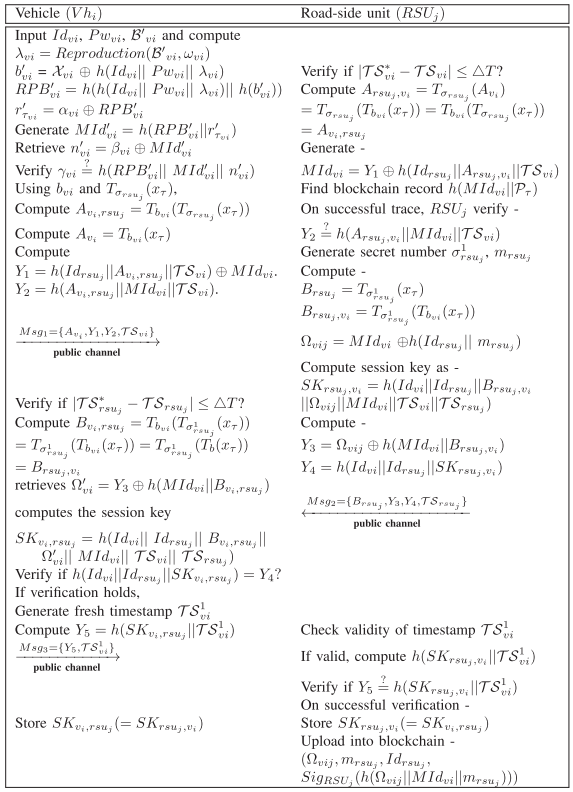


Fig. 5. Authentication between Vh_i and RSU_j.

Vehicle user gives identity, password and biometrics at OBU_{vi} and secret key is computed from the stored $\alpha_{vi}, \beta_{vi}, \gamma_{vi}$ and ω_{vi}.

4) Inter-RSU Vehicle Authentication Handover Using Blockchain: Suppose, a vehicle Vh_i, which is having current session key shared with RSU_j, wants to migrate to a coverage area of another roadside unit RSU_k having id Id_{rsu_k}. For this, Vh_i needs to execute the following steps for the authentication handover.

Step AHVR1:

1) Vh_i generates an 128-bit random number θ_{vi}.
2) Vh_i computes hash values $Y_5 = h(MId_{vi}||\ \Omega_{vij}||\ Id_{rsu_k}||\ \mathcal{TS}^1_{vi}) \oplus \theta_{vi}$ and $Y_6 = h(MId_{vi}||\Omega_{vij}||\theta_{vi})$.
3) Vh_i sends authentication request $\{\Omega_{vij}, Y_5, Y_6, \mathcal{TS}^1_{vi}\}$ to RSU_k though a public channel.

Step AHVR2:

1) RSU_k Verify if $|\mathcal{TS}^{1\ *}_{vi} - \mathcal{TS}^1_{vi}| \leq \triangle T$? Delay more than expected leads to rejection of the request message.
2) Otherwise, RSU_k Retrieves from Blockchain record $(\Omega_{vij}, m_{rsu_j}, Id_{rsu_j}, Sig_{RSU_j}(h(\Omega_{vij}||MId_{vi}||m_{rsu_j})))$. Next, it computes $MId^*_{vi} = \Omega_{vij} \oplus h(Id_{rsu_j}||m_{rsu_j})$ and verify if $h(\Omega_{vij}||\ MId^*_{vi}||m_{rsu_j}) \stackrel{?}{=} h(\Omega_{vij}||\ MId_{vi}||m_{rsu_j})$.
3) If verification holds, RSU_k retrieves $\theta'_{vi} = Y_5 \oplus h(MId^*_{vi}||\Omega_{vij}||Id_{rsu_k}||\mathcal{TS}^1_{vi})$ and verifies if $h(MId_{vi}||\Omega_{vij}||\theta'_{vi}) \stackrel{?}{=} Y_6$. A mismatch leads to the rejection of the handover request message.
4) On success, generate two 128-bit random numbers θ_{rsu_k} and m_{rsu_k}. Next, it computes $\Omega_{vik} = MId_{vi} \oplus h(Id_{rsu_k}||\ m_{rsu_k})$, $Y_7 = h(\Omega_{vik}) \oplus h(\Omega_{vij}||\theta_{vi})$, $Y_8 = h(\Omega_{vik}||MId_{vi}||\theta_{vi}) \oplus \theta_{rsu_k}$.
5) Using these computed parameters, RSU_k computes session key shared with Vh_i as $SK_{rsu_k,v_i} = h(MId_{vi}||\ \Omega_{vik}||\ \theta_{vi}||\ \theta_{rsu_k})$.
6) RSU_k computes $Y_9 = h(SK_{rsu_k,v_i}||\ \mathcal{TS}^1_{rsu_k}))$ and finally sends $\{Y_7, Y_8, Y_9, \mathcal{TS}^1_{rsu_k}\}$ to Vh_i through a secure channel.

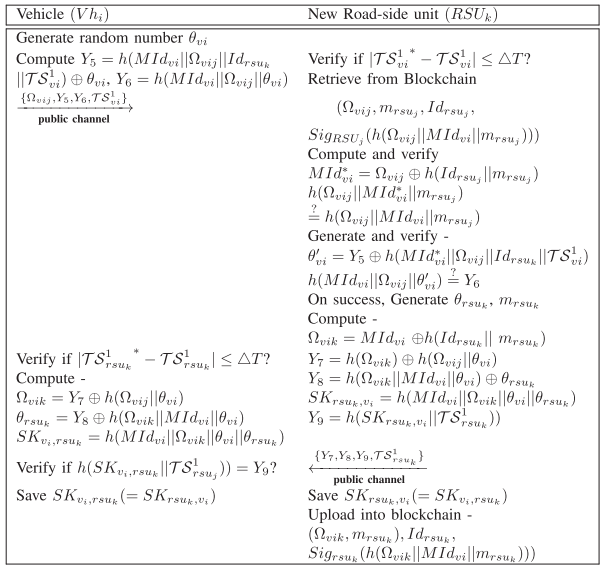


Fig. 6. Authentication handover between Vh_i and new RSU_k.

Step AHVR3:

1) Verify if $|\mathcal{TS}^{1\ *}_{rsu_k} - \mathcal{TS}^1_{rsu_k}| \leq \triangle T$? Otherwise, Vh_i rejects this reply message.
2) Vh_i retrieves $\Omega_{vik} = Y_7 \oplus h(\Omega_{vij}||\theta_{vi})$ and $\theta_{rsu_k} = Y_8 \oplus h(\Omega_{vik}||MId_{vi}||\theta_{vi})$.
3) Using retrieved parameters,Vh_i computes the session key shared with RSU_k as $SK_{v_i,rsu_k} = h(MId_{vi}||\Omega_{vik}||\theta_{vi}||\theta_{rsu_k})$.
4) Finally, Vh_i verify if $h(SK_{v_i,rsu_k}||\mathcal{TS}^1_{rsu_j})) \stackrel{?}{=} Y_9$. If this verification is successful, then both Vh_i and RSU_k save their session keys.

The authentication handover procedure between Vh_i and its new RSU_k is also briefed in Fig. 6 .

Remark 3: It is to be noted that, during either the access control phase or the authentication handover phase, the session key between a vehicle Vh_i and its associated RSU is never written into the blockchain ledger as in [27]. At the end of the successful authentication with Vh_i, RSU_j stores parameters $\{\Omega_{vij}, m_{rsu_j}, Id_{rsu_j}\}$ and uses "Elliptic Curve Digital Signature Algorithm (ECDSA)" [59] to sign and store the hash parameter $Sig_{RSU_j}(h(\Omega_{vij}||MId_{vi}||m_{rsu_j}))$ into the blockchain ledger. Here, Ω_{vij} is created using credentials of both vh_i and RSU_j. As these parameters are stored in the blockchain and signed by RSU_j, it is ensured that they are not tampered with and that the record is stored by entity RSU_j only. Later, when the migrating vehicle Vh_i wants to establish a session key with a new roadside unit RSU_k, RSU_k will use the stored and signed

blockchain parameters to verify if Vh_i is authentic and known to the earlier roadside unit RSU_j. On successful verification, RSU_k exploits the blockchain parameters to create Ω_{vik} and establish a new session key SK_{rsu_k,v_i} with Vh_i. Through this handover mechanism, the repetition of the authentication and key establishment process has been avoided which helps in reducing significantly the communication and computational overheads.

5) Dynamic Vehicle Addition Phase: In a smart city or intelligent transportation system applications, new vehicles are needed to be added into the system quite frequently. Thus the process of inclusion of new vehicles must be simple and efficient. A new vehicle Vh_{new} executes the following steps to get registered and join the system.

Step NVA1: New vehicle user Vh_{new} chooses its unique identity, password as Id_{new}, Pw_{new} and imprints biometrics $\mathcal{B}_{new}$ in its on-board unit (OBU_{new}). Vh_{new} generates $(\lambda_{new}, \omega_{new}) = Generation(\mathcal{B}_{new})$, computes masked password $VPB_{new} = h(Id_{new}||\ Pw_{new}||\ \lambda_{new})$ and delivers (Id_{new}, VPB_{new}) to $\mathcal{TA}$ via a secured channel.

Step NVA2: $\mathcal{TA}$ checks availability of the new vehicle id Id_{new}, and generates a 1024-bit unique secret key b_{new} for Vh_{new}. for Vh_i, $\mathcal{TA}$ chooses an 1024-bit unique number $r_{\tau_{new}}$, and computes $RPB_{new} = h(VPB_{new}||\ h(b_{new}))$, $MId_{new} = h(RPB_{new}||\ r_{\tau_{new}})$.

Step NVA3: $\mathcal{TA}$ computes $\alpha_{new} = RPB_{new} \oplus r_{\tau_{new}}$, generates a random nonce n_{new} and computes $\beta_{new} = MId_{new} \oplus n_{new}$. $\mathcal{TA}$ computes two parameters $\gamma_{new} = h(RPB_{new}||\ MId_{new}||\ n_{new})$ and $\mathcal{X}_{new} = VPB_{new} \oplus b_{new}\mathcal{TA}$ uploads $< h(MId_{vi}||\mathcal{P}_\tau) >$ into the blockchain and loads $\{\alpha_{vi}, \beta_{vi}, \gamma_{vi}, \mathcal{X}_{vi}\}$ in new on-board unit OBU_{new}. Finally, $\mathcal{TA}$ computes a Chebyshev polynomial $T_{b_{new}}(x_\tau)$ and declares it as public key of Vh_{new}.

Step NVA4: Vh_{new} receives OBU_{new} with parameters $(\alpha_{vi}, \beta_{vi}, \gamma_{vi}, \mathcal{X}_{vi})$. Vh_{new} adds $\{\omega_{vi}\}$ into OBU_{new}.

6) Vehicle Revocation Phase: Suppose, RSU_x detects some malicious activity of a registered vehicle, say Vh_a, and decides to revoke that vehicle from the system. RSU_x needs to execute the following steps:

Step VhRev1: RSU_x finds out the unique masked password MId_{va} of vehicle Vh_a. RSU_x can retrieve MId_{va} in Step VRAC3 of Section IV-B3. RSU_x sends MId_{va} to $\mathcal{TA}$ through secure channel and requests to revoke Vh_a.

Step VhRev2: $\mathcal{TA}$ uses its public parameter $\mathcal{P}_\tau$, computes $h(MId_{va}||\mathcal{P}_\tau)$, finds it from blockchain, and deletes record $h(MId_{va}||\mathcal{P}_\tau)$ from blockchain. $\mathcal{TA}$ finds record $< MId_{va}, \mathcal{B}_a >$ from blockchain and saves it into a revocation list (See Step VHR3 of Section IV-B2). As biometric $\mathcal{B}_a$ is immutable, revoked vehicle Vh_a can not re-register into $\mathcal{TA}$, even if it uses new id and password during re-registration.

Remark 4: As mentioned in our threat model (Section III-B), RSUs are not assumed to be a fully trusted entity and may turn hostile to launch privileged insider attacks. Consequently, an adversary RSU can deliver the obtained masked password MId_{va} to an adversary vehicle user (say Vh_r) or other RSU. However, obtaining this masked password MId_{va}, the adversary

Block Header	
Merkle Tree Root	MTR_l
Block Owner	RSU_j
Signer's Public Key	Pub_{RSU_j}
Block Payload (Transactions)	
List of n_{thr} transactions	$\{Trx_i \mid i = 1, 2, \cdots, n_{thr}\}$
Block Signature using ECDSA signature	$Sig_{RSU_j}(Block_l)$

Fig. 7. Construction of a partial block $PartialBlock_l$.

Vh_r and RSUs can not launch any security attack into the system. According to Step VRAC1, a vehicle Vh_i needs to verify if stored $\gamma_{vi} = h(RPB'_{vi}||\ MId'_{vi}||\ n'_{vi})$, where $RPB'_{vi} = h(h(Id_{vi}||\ Pw_{vi}||\ \lambda_{vi})||\ h(b'_{vi}))$. Here, password Pw_{vi} and fuzzy extractor biometric parameter λ_{vi} of vehicle Vh_i are unknown to Vh_r, and thus can not compute RPB'_{vi}. Hence, the authentication process terminates and Vh_r can not launch an impersonation attack. Moreover, the vehicle user Vh_r and RSU can not execute handover authentication. According to phase in Section IV-B4, Vh_r or an RSU needs to compute Ω_{vij}, that contains B_{v_i,rsu_j}, a parameter built on the mutual trust of a Vh_i and RSU_j during authentication setup. So, the illegal attempt of handover authentication fails.

7) Secure Data Aggregation Phase: It is worth noticing that through the authentication procedure between a vehicle Vh_i and it is nearby RSU_j described in Section IV-B3, both Vh_i and RSU_j establish the session key SK_{v_i,rsu_j} for their secret communications. Likewise, using the authentication handover procedure between Vh_i and its new RSU_k provided in Section IV-B4, both Vh_i and RSU_k also establish the session key SK_{v_i,rsu_k} for their secret communications. Thus, the messages passed from the vehicles to their nearby RSUs are securely aggregated using the corresponding session keys. In ITS, there can be two types of messages: a) "Periodic Safety Message (referred in the sequel as Beacon)" and b) "Event-Driven Message (referred to as Emergency Message)" [60]. For instance, RSUs can have secure transactions (messages) Trx_i, once a vehicle accident (either the same vehicle or its nearby neighbor vehicle(s)) has been detected [49]. It is also noted that $\mathcal{TA}$ delivers information about the registered vehicles, like randomized masked passwords, 128-bit random number to the blockchain during our access control phase and inter-RSU vehicle authentication handover phase so that RSU can check the authenticity of a particular vehicle. This information is from the transactions. Other transactions (messages) may be traffic-related messages.

8) Blockchain Implementation Phase: Once the transactions (messages) are securely received by an RSU_j residing in a group of vehicles Vh_i, RSU_j starts forming a partial block, say $PartialBlock_l$ with the n_{thr} transactions Trx_i. Note that here we have shown a block with respect to a public blockchain. In ITS, some transactions may be private and confidential, and those transactions may be encrypted with the public key of the block owner, i.e., Pub_{RSU_j}. Thus, a consortium block may be best suited for an ITS environment. Once a partial block shown in Fig. 7 is constructed, it is then forwarded to a cloud server

Block Header	
Block Version	$BVer_l$
Previous Block Hash	$PBHash_l$
Block Timestamp	BTS_l
Merkle Tree Root	MTR_l
Block Owner	RSU_j
Signer's Public Key	Pub_{RSU_j}
Block Payload (Transactions)	
List of n_{thr} transactions	$\{Trx_i \mid i = 1, 2, \cdots, n_{thr}\}$
Block Signature using ECDSA signature	$Sig_{RSU_j}(Block_l)$
Current Block Hash	$CBHash_l$

Fig. 8. Construction of a full block $Block_l$.

node in the blockchain center. After receiving the partial block, the in-charge cloud server node will proceed to convert it to a full block. If we consider such a full block shown in Fig. 8, it contains the following information (fields):

- *Block Version* $(BVer_l)$*:* It is the unique serial number to a block, $Block_l$.
- *Previous Block Hash* $(PBHash_l)$*:* It denotes the hash value of $Block_l$'s previous block.
- *Block Timestamp* (BTS_l)*:* It represents the block formation timestamp.
- *Merkle Tree Root* (MTR_i)*:* It is computed using all the n_{thr} transactions Trx_i present in the constructed partial block $PartialBlock_l$, which contains the hash value of all the transactions indirectly.
- *Block Owner* (RSU_j)*:* Here, the in-charge RSU, RSU_j will act as the owner of the block.
- *Signer's Public Key* (Pub_{RSU_j}): The public key Pub_{RSU_j} of the signer (in this case, RSU_j) will be used to verify the signature of the block by any verifier, even during the consensus time. Note that each RSU_j will have their own ECC-based private key $pr_{RSU_j} \in E_p(\alpha, \beta)$ and public key $Pub_{RSU_j} = pr_{RSU_j} \cdot G$.
- *Transactions* $\{Trx_i\}$: The transactions can be "Periodic Safety Message" or "Event Driven Message (Emergency Message)".
- *Block Signature* $(Sig_{RSU_j}(Block_l))$: The block signature on all the n_{thr} transactions Trx_i present in the block is computed using the "Elliptic Curve Digital Signature Algorithm (ECDSA)" [59].
- *Current Block Hash* $(CBHash_l)$*:* It is the hash of all the fields contained in the block header as well as the block payload.

As in [49], we also apply the voting-based "Practical Byzantine Fault Tolerance (PBFT)" consensus algorithm [61] for mining the constructed full block, $Block_l$. The inputs in the consensus algorithm will be the following: 1) $Block_l = \{$Block Header, Block Payload, $CBHash_l\}$, 2) the "private-public key pairs $(pr_{CS_w}, Pub_{CS_w} = pr_{CS_w} \cdot G)$ for all the cloud server nodes CS_w in the Peer-to-Peer (P2P) CS blockchain network", and 3) f_{cs}: the number of "faulty cloud server nodes in the P2P CS blockchain network". During the consensus, once the block $Block_l$ is successfully validated, it will be added to the blockchain.

TABLE III
DIFFERENT ORACLE QUERIES AND THEIR DESCRIPTIONS

Query	**Description/purpose**
Send$(\mathcal{P}^t, m)$	It enables $\mathcal{A}$ to send request message m to $\mathcal{P}^t$ and $\mathcal{P}^t$ replies accordingly
Corrupt(Vh_i, a)	Depending on a, $\mathcal{A}$ can obtain biometric and password of Vh_i
Test$(\mathcal{P}^t)$	$\mathcal{A}$ requests $\mathcal{P}^t$ for the session key SK, $\mathcal{P}^t$ replies probabilistically on outcome of a flipped coin b
Execute(Vh_i, RSU_j)	It enables $\mathcal{A}$ to eavesdrop the messages communicated between Vh_i and RSU_j
Reveal$(\mathcal{P}^t)$	It enables $\mathcal{A}$ to obtain the session key SK generated between $\mathcal{P}^t$ and its partner

TABLE IV
SYMBOLS USED IN THE REAL-OR-RANDOM (ROR) MODEL

Symbol	**Description**
q_H	Total number of hash H oracle queries execution
q_s	Total number of $Send$ oracle queries execution
q_e	Total number of $Execute$ oracle queries execution
l_H	Length of hash output string
l_r	Length of random number string
l_b	Length of user biometric key
ε_{bm}	Probability of false positive in biometrics
$\mathcal{D}$	Password dictionary space with size |D|
L_H	List that stores output of hash H oracle query
L_A	List that records random oracle outputs
L_T	List recording message transcripts between Vh_i and RSU_j

V. SECURITY ANALYSIS

A. Formal Security Analysis Under ROR Model

In this subsection, we first provide the formal security of the proposed BACHP-IoV through a widely used Real-or-Random (ROR) model [55]. Note that the ROR model differs from the traditional random oracle model.

Under a real attack scenario model, an adversary $\mathcal{A}$ makes several queries. Table III contains a brief description of the various oracle queries used for this formal proof. Here, $\mathcal{A}$ interacts with the tth instance of an executing participant (Vh_i or RSU_j), known as $\mathcal{P}^t$. Table IV contains all symbols and notations used for this formal proof.

Definition 1 (Semantic security): Let $Adv_{\mathscr{P}}^{BACHP}$ refer to the advantage function of an adversary $\mathcal{A}$, that aims to break the semantic security of BACHP-IoV in polynomial time through the guessing of a correct bit b'. Then the advantage function is defined by $Adv_{\mathscr{P}}^{BACHP-IoV} = |2Pr[b = b'] - 1|$.

Definition 2: BACHP-IoV is semantically secure if the advantage function $Adv_{\mathscr{P}}^{BACHP}$ is only negligibly greater than $\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\}$, where parameters q_s, $|\mathcal{D}|$ and l_b denotes their conventional meaning as tabulated in Table IV.

Definition 3: The advantage probability $Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}})$ of the Chaotic map-based discrete logarithm problem (CMDLP) is negligible for any adversary $\mathcal{A}$ with execution time $t_{\mathcal{A}}$, that is, $Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}) \leq \epsilon$, for a sufficiently small $\epsilon > 0$.

Theorem 1: Let $Adv_{\mathscr{P}}^{BACHP}$ be the polynomial time bounded advantage function for $\mathcal{A}$ on breaking the semantic security of BACHP-IoV in time $t_{\mathcal{A}}$. Then, $Adv_{\mathscr{P}}^{BACHP} \leq \frac{q_H^2 + 22q_H}{2^{l_H}} + \frac{(q_s+q_e)^2+4q_s}{2^{l_r}} + 2\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} + 4q_H(1 + (q_s + q_e)^2)Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}})$, where $q_H, q_s, q_e, \mathcal{D}, l_H, l_r, l_b$ and ε_{bm} have their usual meanings as tabulated in Table IV.

Proof: To begin with the proof, we consider five games G_i $(i = 0, 1, 2, 3, 4)$ where in each game G_i, $\mathcal{A}$ executes $Test$ query

to guess a correct bit b through the $Test$ query. We define the event and the corresponding probability as S_i and $Pr[S_i]$, respectively. Due to the number of pages limitation, we have shifted the details of each game G_i and their analysis in the supplementary material. ■

B. Informal Security Analysis

In this subsection, through a set of propositions, we present the strength of the proposed BACHP-IoV on resistance to various security attacks.

1) Replay Attack: During access control process between Vh_i and RSU_j, BACHP-IoV first sends authentication request message $Msg_1 = \{A_{v_i}, Y_1, Y_2, \mathcal{TS}_{vi}\}$ to RSU_j though a public channel. On receiving Msg_1 at time $\mathcal{TS}^*_{vi}$, RSU_j verifies if $|\mathcal{TS}^*_{vi} - \mathcal{TS}_{vi}| \leq \triangle T$? If the message transmission delay is more than the threshold value $\triangle T$, RSU_j discards Msg_1. Further, an adversary $\mathcal{A}$ can not change or replay any parameter as RSU_j generates $h(MId_{vi}||\mathcal{P}_\tau)$ and checks if received $Y_2 = h(A_{rsu_j,v_i}||MId_{vi}||\mathcal{TS}_{vi})$. If verification fails, RSU_j terminates the authentication process. In reply, RSU_j sends $Msg_2 = \{B_{rsu_j}, Y_3, Y_4, \mathcal{TS}_{rsu_j}\}$ to Vh_i as authentication reply message. Vh_i receives RSU_j reply message Msg_2 and verifies if $|\mathcal{TS}^*_{rsu_j} - \mathcal{TS}_{rsu_j}| \leq \triangle T$? If the message transmission delay is more than the threshold value $\triangle T$, Vh_i discards Msg_2. Finally, RSU_j computes SK_{v_i,rsu_j} compare the hash value $h(Id_{vi}||Id_{rsu_j}||SK_{v_i,rsu_j}) = Y_4$? A mismatch leads to the termination of the authentication process. As a result, BACHP-IoV resists all replay attacks.

2) Man-in-The-Middle (MiTM) Attack: Suppose, adversary $\mathcal{A}$ intercepts authentication request message Msg_1 and sends its own malicious message $Msg'_1 = \{A'_{v_i}, Y'_1, Y'_2, \mathcal{TS}'_{vi}\}$ to RSU_j. Using these modified parameters, RSU_j computes $MId'_{vi} = Y'_1 \oplus h(Id_{rsu_j}||A^*_{rsu_j,v_i}||\mathcal{TS}'_{vi})$. However, RSU_j searches the blockchain ledger and fails to find the stored record $h(MId'_{vi}||\mathcal{P}_\tau)$ and the authentication process terminates. Hence, BACHP-IoV defends against MiTM attack.

3) OBU Physical Capture Attack: As OBU_{vi} is not hardware tamper-proof, it cannot be considered a fully trusted unit. Using the power analysis attack, $\mathcal{A}$ can extract all the stored credentials $\{\alpha_{vi}, \beta_{vi}, \gamma_{vi}\}$ and $\mathcal{X}_{vi}$ from OBU_{vi} of Vh_i. However, as explained in smart vehicle registration phase, $\alpha_{vi} = RPB_{vi} \oplus r_{\tau_{vi}}$, $\beta_{vi} = MId_{vi} \oplus n_{vi}$, $\gamma_{vi} = h(RPB_{vi}||MId_{vi}||n_{vi})$, and $\mathcal{X}_{vi} = VPB_{vi} \oplus b_{vi} = h(Id_{vi}||Pw_{vi}||\lambda_{vi}) \oplus b_{vi}$. However, due to one-way property of cryptographic hash function $h(\cdot)$, $\mathcal{A}$ cannot retrieve identity Id_{vi}, password Pw_{vi}, or biometrics $\mathcal{B}_{vi}$ from the captured OBU_{vi}. Hence, BACHP-IoV is resilient against OBU physical capture attack.

4) Privileged Insider Attack: The administrator or privileged insider of a $\mathcal{TA}$ could turn into an adversary $\mathcal{A}$ and might try to launch a privileged insider attack. However, BACHP-IoV is secure against this attack. When a new smart vehicle Vh_i needs to register, he/she delivers $(Id_{vi}, \mathcal{B}_{vi}, VPB_{vi})$ to $\mathcal{TA}$ via a secured channel. Here, $VPB_{vi} = h(Id_{vi}||Pw_{vi}||\lambda_{vi})$. So, $\mathcal{A}$ cannot extract password Pw_{vi} of Vh_i. Moreover, at the end of Vh_i and RSU_j registration process, $\mathcal{TA}$ deletes secret parameters b_{vi}, σ_{rsu_j} and biometric $\mathcal{B}_{vi}$ from its memory. So, a hostile privileged insider of $\mathcal{TA}$ cannot obtain any secret credentials from its memory. Therefore, BACHP-IoV defends any possible privileged insider attack.

5) Ephemeral Secret-Key Leakage (ESL) Attacks: RSU_j computes session key shared with Vh_i as $SK_{rsu_j,v_i} = h(Id_{vi}||Id_{rsu_j}||B_{rsu_j,v_i}||\Omega_{vij}||MId_{vi}||\mathcal{TS}_{vi}||\mathcal{TS}_{rsu_j})$, where $\Omega_{vij} = MId_{vi} \oplus h(Id_{rsu_j}||m_{rsu_j})$. Here, m_{rsu_j} is a 128-bit random nonce or a short-term secret. Further, element $B_{rsu_j,v_i} = T_{\sigma^1_{rsu_j}}(T_{b_{vi}}(x_\tau))$, where $\sigma^1_{rsu_j}$ is a secret key which is randomly generated by RSU_j separately on every login session. Based on the CK-adversary model defined in the threat model (see Section III-B), $\mathcal{A}$ can derive the session key SK_{rsu_j,v_i} only if both the long-term as well as ephemeral secrets like $\sigma^1_{rsu_j}$ and m_{rsu_j} are compromised, which is practically an infeasible task. Therefore, under the CK-adversary model, BACHP-IoV withstands ESL attacks.

6) Known Key Secrecy: Due to the use of unique and random long and short tern secrets, the session keys like SK_{rsu_j,v_i} and SK_{v_i,rsu_j} are distinct in every session. So, compromise of SK_{rsu_j,v_i} or SK_{v_i,rsu_j} will not help to build or compromise other session keys over future as well as past sessions. This concludes BACHP-IoV is resilient against "session-temporary information attack" as well as it preserves perfect forward and backward secrecy. As stated in another way, BACHP-IoV is secure against known-key secrecy.

7) Online/offline Password-Guessing Attacks: During access control phase, Vh_i sends authentication request message $Msg_1 = \{A_{v_i}, Y_1, Y_2, \mathcal{TS}_{vi}\}$ to RSU_j though a public channel. Due to chaotic maps-based discrete logarithm problem (CMDLP) and one-way property of hash function, it is computationally infeasible to compute vehicle password Pw_{vi} From Msg_1 parameters A_{v_i}, Y_1, Y_2 and $\mathcal{TS}_{vi}$). In reply, RSU_j sends $Msg_2 = \{B_{rsu_j}, Y_3, Y_4, \mathcal{TS}_{rsu_j}\}$ to Vh_i though a public channel. Following similar logic, $\mathcal{A}$ cannot compute Pw_{vi} from Msg_2 parameters either. Hence, BACHP-IoV can defend against possible online password-guessing attacks.

BACHP-IoV is also safe from offline password-guessing attacks. $\mathcal{TA}$ loads parameters $(\alpha_{vi}, \beta_{vi}, \gamma_{vi}, \mathcal{X}_{vi})$ into the OBU_{vi} of Vh_i. Here, $\alpha_{vi} = RPB_{vi} \oplus r_{\tau_{vi}}$, $\beta_{vi} = MId_{vi} \oplus n_{vi}$, $\gamma_{vi} = h(RPB_{vi}||MId_{vi}||n_{vi})$ and $\mathcal{X}_{vi} = VPB_{vi} \oplus b_{vi}$. From none of these parameters, Pw_{vi} can be generated. Moreover, RSU_j and $\mathcal{TA}$ do not store any secret credential which leads to disclosure of Pw_{vi} of Vh_i. As a result, BACHP-IoV is resilient to all offline password-guessing attacks.

VI. Formal Security Verification Using ProVerif

This subsection presents code and simulation results for an analysis of the security verification of the proposed access control scheme using the ProVerif2.03 simulation tool [62]. The tool is based on applied pi-calculus and can be used to verify whether an attacker can attack the session key [62], [63]. The tool has the ability of an attack reconstruction, i.e, when a property cannot be proved, an execution trace that falsifies the desired property is constructed.

ProVerif simulation of the proposed scheme requires the declaration of various channels, constants, free variables, equations,

TABLE V
AVERAGE EXECUTION TIME FOR CRYPTO PRIMITIVES

Operation	Description of operation	Computation time (in ms)
T_{mul}	Elliptic curve point multiplication	0.674
T_{add}	Elliptic curve point addition	0.002
T_{siggen}	ECC signature generation	0.729
T_{sigver}	ECC signature verification	1.405
T_{bp}^{mul}	Multiplication on pairing-based group	0.473
T_{bp}^{add}	Addition on pairing-based group	0.003
T_{bp}	Bilinear pairing operation	4.716
T_H	Hash function	0.006
T_{exp}	Modular exponentiation	0.072
T_{enc}	Symmetric key encryption using AES	0.003
T_{dec}	Symmetric key decryption using AES	0.003
T_{ch}	Chebyshev polynomial operation	0.225
T_{mtp}	Map-to-point	0.114
T_{FE}	Fuzzy extractor operation	0.674
T_{XOR}	XOR operation	negligible

TABLE VI
COMPARATIVE ANALYSIS ON COMPUTATIONAL COSTS

Scheme	Initial Authentication		Handover Authentication	
	Vh_i	RSU_j	Vh_i	(RSU_j+RSU_k)
[66]	$2T_{mul} + 8T_H + T_{enc} \approx 1.395$ ms	$2T_{mul} + T_H + 3T_{enc} + T_{dec} + T_{sigver} \approx 2.7705$ ms	$3T_{mul} + 2T_H + 2Tenc + Tenc \approx 2.036$ ms	$2T_{mul} + 3T_H \approx 1.4075$ ms
[37]	$3T_{exp} + T_{bp}^{mul} + T_{bp} \approx 5.4049$ ms	$5T_{exp} + T_{bp}^{mul} + T_{bp} \approx 5.5489$ ms	$T_{bp} + T_{exp} + T_{bp}^{mul} \approx 5.2609$ ms	$T_{bp} + 4T_{exp} + T_{bp}^{mul} \approx 5.4769$ ms
[27]	$4T_{mul} + 10T_H \approx 2.751$ ms	$3T_{mul} + 8T_H + T_{siggen} \approx 2.795$ ms	$6T_H \approx 0.033$ ms	$T_{sigver} + T_{siggen} + 9T_H \approx 2.1835$ ms
BACHP-IoV	$10T_H + 3T_{ch} + T_{FE} \approx 1.4028$ ms	$9T_H + 3T_{ch} + T_{siggen} \approx 1.4523$ ms	$6T_H \approx 0.033$ ms	$9T_H + 2T_{sigver} + T_{siggen} \approx 3.5885$ ms

and events, which are provided in the supplementary material. Moreover, the ProVerif source codes for the process of Vh_i registration, login, and authentication with the roadside unit, and RSU_j registration and authentication setup with Vh_i are provided in the supplementary material. From the ProVerif simulation results provided in the supplementary material, the following observations are drawn from the results:

- RESULT not attacker(SKvhrsu[]) is true.
- RESULT not attacker(SKrsuvh[]) is true.
- RESULT inj-event (UserAuth(id)) ==> inj-event (UserStart(id)) is true.

VII. PERFORMANCE COMPARISON

In this section, we analyze the performance comparison of our proposed handover authentication scheme, with other existing relevant protocols [27], [37], [64].

A. Comparison on Computation Costs

We simulate the computational cost of each operation in Table V, using MIRACL Library [65]. We perform experiments in a desktop environment. Desktop configuration is Memory 7.7 GiB, Intel quad Core i7-8565 U, Linux Ubuntu 20.04.4 desktop-amd64 operating system. Table VI summarizes the total computational cost of our protocol and the existing relevant handover authentication protocol. Vehicle denoted by V_i, RSU_s that authenticate with V_i at initial phase denoted by RSU_j and authenticate at handover authentication phase is denoted by RSU_k. The computations values of our proposed schemes are 2.8551 ms in the case of the initial authentication phase, and 3.5885 ms for the handover authentication phase, whereas the computation cost of Xu et al.'s scheme [64] is 4.1655 ms in initial authentication and 3.4435 ms in handover authentication, Wang et al.'s scheme [37] needs 10.9538 ms and 15.2198 ms in initial and handover authentication respectively, Son et al.'s scheme [27] requires 5.546 ms in initial authentication and 2.2165 ms in the handover authentication scheme. It is clear from Table VI that our proposed scheme requires very less time compared to other handover authentication protocols.

TABLE VII
COMPARATIVE ANALYSIS ON COMMUNICATION COSTS

Scheme	Initial Authentication		Handover Authentication	
	# messages	Total Cost (in bits)	# messages	Total Cost (in bits)
Xu et al. [66]	6	4128	5	3552
Wang et al. [37]	1	1056	3	2048
Son et al. [27]	2	1408	2	1600
BACHP-IoV	3	2336	2	1600

B. Comparison on Communication Costs

Here we consider two cases. one for initial authentication and another for handover authentication. We consider 1024 bits for multiplication operation on bilinear pairing, 256 bits for a one-way hash function, 320 bits for ECC point multiplication, and 32 bits for random numbers or timestamps. In BACHP-IoV, the initial authentication phase uses three messages. $Msg_1 = \{A_{v_i}, Y_1, Y_2, \mathcal{TS}_{vi}\}$ needs 512 bits, $Msg_2 = \{B_{rsu_j}, Y_3, Y_4, \mathcal{TS}_{rsu_j}\}$ needs 512 bits, and $Msg_3 = \{Y_5, \mathcal{TS}_{vi}^1\}$ needs 192 bits. The total communication cost of the initial authentication phase is 2336 bits. The BACHP-IoV handover authentication phase uses two messages. $HMsg1 = \{\Omega_{vij}, Y_5, Y_6, \mathcal{TS}_{vi}^1\}$ needs 512 bits, $HMsg2 = \{Y_7, Y_8, Y_9, \mathcal{TS}_{rsu_k}^1\}$ needs 512 bits. The total communication cost of the handover authentication phase is 1600 bits. Table VII consisting the comparison of the communication cost of our proposed model and other existing models. The communication cost of Xu et al. [64] are 4128 bits and 3552 bits for initial and handover authentication respectively, Wang et al. [37] has 1056 bits and 2048 bits, Son et al. has 1408 bits and 1600 bits for initial and handover authentication respectively [27]. Wang et al. [37] has a lesser communication cost, but it has a large computation cost. Son et al. [27] has lesser communication cost than the proposed BACHP-IoV, but this scheme suffers from ESL attack under the CK-adversary model.

C. Security and Functionality Attributes Comparison

Based on various security and functionality features, Table VIII presents a detailed comparison of BACHP-IoV with schemes proposed by Xu et al. [64], Wang et al. [37] and Son et al. [27]. We compare on a total of fourteen attributes (F_1 to F_{14}). The results tabulated in Table VIII show that the proposed BACHP-IoV is more secure and supports more functionality than the existing schemes.

TABLE VIII
SECURITY AND FUNCTIONALITY FEATURES COMPARISON

Attributes	[66]	[37]	[27]	BACHP-IoV
F_1	✓	✓	✓	✓
F_2	✓	✓	✓	✓
F_3	✓	✓	✓	✓
F_4	-	-	✓	✓
F_5	-	-	✓	✓
F_6	-	-	✓	✓
F_7	X	X	✓	✓
F_8	X	X	✓	✓
F_9	✓	✓	✓	✓
F_{10}	✓	✓	✓	✓
F_{11}	X	✓	✓	✓
F_{12}	✓	-	X	✓
F_{13}	X	X	X	✓
F_{14}	X	X	X	✓
F_{15}	X	X	X	✓

F_1: "Resistance to replay attack"; F_2: "Resistance to impersonation attack"; F_3: "Resistance to session key disclosure attack"; F_4: "Resistance to privileged-insider attack"; F_5: Preservation of perfect forward secrecy; F_6: "Resistance to ephemeral key leakage attack";F_7: "Support of RSU fault tolerance"; F_8: "Support of handover integrity"; F_9: "Preservation of anonymity"; F_{10}: "Preservation of untraceability"; F_{11}: "Support of decentralization"; F_{12}: "Support of password and biometric update"; F_{13}: "Supports dynamic vehicle addition"; F_{14}: "Low computation overhead"; F_{15}: "Resistance to de-synchronization attack".

TABLE IX
DIFFERENT SCENARIOS USED IN SIMULATION STUDY

Part 1: Constant number of clusters		Part 2: Constant number of vehicles per cluster	
Case 1	40 vehicles, 4 clusters	Case 1	90 vehicles, 6 clusters
Case 2	60 vehicles, 4 clusters	Case 2	120 vehicles, 8 clusters
Case 3	80 vehicles, 4 clusters	Case 3	135 vehicles, 9 clusters
Case 4	100 vehicles, 4 clusters		

VIII. PRACTICAL IMPLEMENTATION

A. NS3 Simulation

Several network parameters are computed to analyze the network performance during the V2I mutual authentication and handover phases described in Section IV-B3. During the simulation study, we conducted experiments with varying numbers of clusters and traffic congestion.

1) Simulation Parameters and Environment: NS-3.36 simulation tool [66] was used to perform the simulation on the Ubuntu 20.04.4 LTS platform. The wireless protocol IEEE 802.11a was used with the rate control algorithm, ConstantRateWifiManager. The widely accepted "Ad-hoc On-demand Distance Vector (AODV)" was used as the routing protocol. For all other parameters, standard values were taken related to vehicular communications.

The RSUs are arranged in a rectangular grid, each being 50 units apart. A unit equals 20 meters. Each cluster has one RSU at its center. Vehicles are randomly and uniformly positioned in the rectangular grid. They are considered to be moving with a constant mobility of 60 km/hr. A total of 14 experiments were conducted. In Part 1, there are 4 cases with varying numbers of vehicles considering 4 clusters. In Part 2, there are 3 cases with varying numbers of clusters considering 15 vehicles per cluster. For each of these 7 cases, we have 2 scenarios (as shown in Table IX):

Scenario 1: Vehicles are static and authenticate with RSU of the cluster they are present in. The simulation runs for 20 s.

Scenario 2: Vehicles have constant mobility and authenticate with nearest RSU. The handover happens only when vehicles move from one cluster to another. Scenario 2 is a continuation of Scenario 1. The entire simulation was run for 300 s.

2) Discussions on Simulation Results: We discuss below the results obtained from experiments. The NS3 FlowMonitor module is used for the calculation of each of these metrics. The simulation results are provided in the supplementary material.

∗ *Impact on Packet Delivery Ratio (PDR):* PDR is measured as the ratio of the total number of delivered packets to the total number of sent packets. Here, we consider the sum of all packet flows in the network throughout the simulation. The high packet delivery ratio indicates that our scheme is reliable. For Scenario 1, PDR is at 100% when vehicles are few and decrease with more vehicles as a large number of packets get exchanged in a short period of time. However, for Scenario 2, the PDR seems quite stable varying only slightly between 88.7% – 91.2%. This can be explained as at any instant, only a few vehicles will change their cluster. A similar trend is observed with varying numbers of clusters.

∗ *Impact on Throughput:* Throughput is another network parameter that indicates the rate of successful message delivery in the network. A low throughput indicates that our scheme exerts only a little load on the network. Throughput is low for Scenario 1 and moderate for Scenario 2. With a higher number of clusters, the throughput increases steadily.

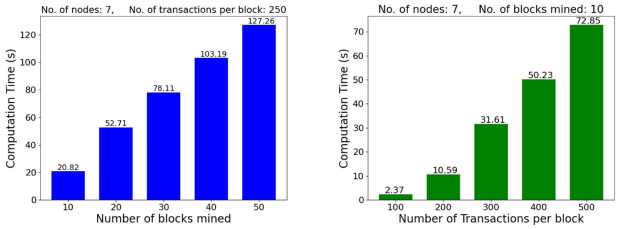
∗ *Impact on End-to-End Delay (EED):* End-to-End Delay is measured as the time taken by a packet transmitted in the network to travel from the source node to the destination node. The end-to-end delay is quite low, less than a split second. This demonstrates that our scheme is robust for quick authentication of vehicles moving at high speeds on roads. The trend of EED resembles that of PDR. It increases steadily for Scenario 1 and remains stable for Scenario 2.

∗ *Impact on Packet Delay Variation Rate (PDV):* This is measured as the mean of differences in end-to-end delay for all the packets in the network. It may also be referred to as packet jitter. The PDV increases with a higher number of vehicles due to network congestion as demonstrated in the supplementary material. This means that the EED is consistent, and it demonstrates the reliability of the proposed scheme.

B. Blockchain Simulation

Here, we describe the simulation of our proposed scheme within a blockchain network. The simulation is performed on a Ubuntu 20.04.5 system with processor Intel Core i7-8550 U CPU @ 1.80 GHz having 4 cores and 8 threads.

We have used the Hyperledger Sawtooth blockchain platform [67] for simulating our distributed ledger application. There are 7 nodes in our blockchain network, each consisting of *Validator, REST API, Consensus Engine, and Transaction Processors*. All nodes have the same set of transaction processors for validating the transactions that are submitted through the REST API. In our simulation, we have adopted the widely used



(a) (b)

Fig. 9. Blockchain simulation results, with variation across (a) number of blocks mined (b) TPB (transactions per block).

"Practical Byzantine Fault Tolerance (PBFT)" algorithm [61] for ensuring consensus between nodes.

There are the following types of data that need to be stored on and retrieved from the blockchain. These have been described in our scheme, such as 1) $h(MId_{vi}||\mathcal{P}_\tau)$, 2) $(\Omega_{vij}, m_{rsu_j}, Id_{rsu_j}, Sig_{RSU_j}(h(\Omega_{vij}||MId_{vi}||m_{rsu_j})))$, 3) $< MId_{vi}, \mathcal{B}_{vi} >$, and 4) the real-time data from the vehicles, "Periodic Safety Message" or "Event Driven Message (Emergency Message)". We create a custom transaction handler for the transactions containing data in the above formats. Transactions are wrapped in batches as atomic units of state change. These batches are submitted to the validators at 3-second intervals to prevent overcrowding. This delay is accounted for in our results. To analyze the performance of our blockchain network, we consider the following two cases:

Case 1: In Fig. 9(a), the number of transactions per block is fixed at 250. For 50 blocks, we calculate the computation time from the instant at which the first block is created to when the last block is added to the blockchain. From the figure, we can see that as the number of blocks mined increases, the mining time increases linearly.

Case 2: In Fig. 9(b), we vary the tpb (transactions per block) from 100 to 500, and measure the mining time for 10 blocks. It can be seen from the figure that this mining time increases linearly with the tpb.

IX. Conclusion

In this article, we design a new blockchain-based access control scheme with a handover policy for an IoV-enabled ITS that is not only efficient in communication and computation, but also provides superior security and more functionality attributes as compared to other competing schemes. The proposed scheme allows for both dynamic vehicle addition and vehicle revocation phases. The stored records in the blockchain help in the inter-RSU vehicle authentication handover process. The practical demonstration of the proposed scheme using the NS-3 simulation shows the impact on various network performance parameters. In addition, through the blockchain simulation, we have also shown the computational time needed for the proposed scheme by varying the number of transactions in blocks and the number of blocks mined in the blockchain network. Future research work includes developing a real testbed experiment for implementing the proposed scheme.

Acknowledgment

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback.

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, winter 2019.

[3] L. F. Herrera-Quintero, J. C. Vega-Alfonso, K. B. A. Banse, and E. Carrillo 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, Summer 2018.

[4] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[5] 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.

[6] R. Gasmi and M. Aliouat, "Vehicular ad hoc NETworks versus internet of vehicles - A comparative view," in *Proc. Int. Conf. Netw. Adv. Syst.*, 2019, pp. 1–6.

[7] A. Vangala, S. Roy, and A. K. Das, "Blockchain-based lightweight authentication protocol for iot-enabled smart agriculture," in *Proc. IEEE Int. Conf. Cyber- Phys. Social Intell.*, 2022, pp. 110–115.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] M.-T. Zhou, F.-G. Shen, T.-F. Ren, and X.-Y. Feng, "Blockchain-based volunteer edge cloud for IoT applications," in *Proc. IEEE 93rd Veh. Technol. Conf.*, 2021, pp. 1–6.

[13] K.-K. R. Choo, R. Chaudhary, A. Jindal, G. Aujla, S. Aggarwal, and N. Kumar, "BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system," *Comput. Secur.*, vol. 85, pp. 288–299, 2019.

[14] S. Chatterjee and S. Roy, "An efficient dynamic access control scheme for distributed wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 27, no. 1, pp. 1–18, 2018.

[15] J. Lee, S. Yu, M. Kim, Y. Park, and A. K. Das, "On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks," *IEEE Access*, vol. 8, pp. 107046–107062, 2020.

[16] L. Vishwakarma, A. Nahar, and D. Das, "LBSV: Lightweight blockchain security protocol for secure storage and communication in SDN-Enabled IoV," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 5983–5994, Jun. 2022.

[17] F. Lin, S. Xia, J. Qi, C. Tang, Z. Zheng, and X. Yu, "A parking sharing network over blockchain with proof-of-planned-behavior consensus protocol," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8124–8136, Aug. 2022.

[18] X. Zhou, M. Luo, P. Vijayakumar, C. Peng, and D. He, "Efficient certificateless conditional privacy-preserving authentication for VANETs," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7863–7875, Jul. 2022.

[19] I. Ali, Y. Chen, N. Ullah, R. Kumar, and W. He, "An efficient and provably secure ECC-Based conditional privacy-preserving authentication for vehicle-to-vehicle communication in VANETs," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1278–1291, Feb. 2021.

[20] H. Zhong, L. Pan, Q. Zhang, and J. Cui, "A new message authentication scheme for multiple devices in intelligent connected vehicles based on edge computing," *IEEE Access*, vol. 7, pp. 108211–108222, 2019.

[21] M. Roeschlin, C. Vaas, K. B. Rasmussen, and I. Martinovic, "Bionyms: Driver-centric message authentication using biometric measurements," in *Proc. IEEE Veh. Netw. Conf.*, 2018, pp. 1–8.

[22] "Advantages and disadvantages of biometrics," Accessed: Nov., 2022. [Online]. Available: https://www.miteksystems.com/blog/advantages-and-disadvantages-of-biometrics

[23] "Fingerprint recognition for the car: Use cases and design considerations," Accessed: Nov., 2022. [Online]. Available: https://www.electronicdesign.com/markets/automotive/article/21119162/fingerprint-recognition-for-the-car-use-cases-and-design-considerations

[24] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1–17, 2008.

[25] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Adv. Cryptol. - EUROCRYPT*, 2004, pp. 523–540.

[26] K. Simoens, J. Bringer, H. Chabanne, and S. Seys, "A framework for analyzing template security and privacy in biometric authentication systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 2, pp. 833–841, Apr. 2012.

[27] S. Son, J. Lee, Y. Park, Y. Park, and A. K. Das, "Design of blockchain-based lightweight V2I handover authentication protocol for VANET," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1346–1358, May/Jun. 2022.

[28] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1832–1843, Dec. 2017.

[29] X. Liu, H. Huang, F. Xiao, and Z. Ma, "A blockchain-based trust management with conditional privacy-preserving announcement scheme for vanets," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4101–4112, May 2020.

[30] P. Bagga, A. K. Sutrala, A. K. Das, and P. Vijayakumar, "Blockchain-based batch authentication protocol for internet of vehicles," *J. Syst. Archit.*, vol. 113, 2021, Art. no. 101877.

[31] A. Vangala, A. K. Das, A. Mitra, S. K. Das, and Y. Park, "Blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural IoT networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 904–919, 2023.

[32] A. Mitra, B. Bera, A. K. Das, S. S. Jamal, and I. You, "Impact on blockchain-based AI/ML-enabled Big Data analytics for cognitive Internet of Things environment," *Comput. Commun.*, vol. 197, pp. 173–185, 2023.

[33] H. Tan and I. Chung, "Secure authentication and key management with blockchain in vanets," *IEEE Access*, vol. 8, pp. 2482–2498, 2020.

[34] S. A. Chaudhry, "Comments on a secure, privacy-preserving, and lightweight authentication scheme for VANETs," *IEEE Sensors J.*, vol. 22, no. 13, pp. 13763–13766, Jul. 2022.

[35] T. Nandy et al., "A secure, privacy-preserving, and lightweight authentication scheme for VANETs," *IEEE Sensors J.*, vol. 21, no. 18, pp. 20998–21011, Sep. 2021.

[36] Q. Xie, P. Zheng, Z. Ding, X. Tan, and B. Hu, "Provable secure and lightweight vehicle message broadcasting authentication protocol with privacy protection for VANETs," *Secur. Commun. Netw.*, vol. 2022, 2022, Art. no. 3372489.

[37] C. Wang, J. Shen, J.-F. Lai, and J. Liu, "B-TSCA: Blockchain assisted trustworthiness scalable computation for V2I authentication in VANETs," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1386–1396, Jul.–Sep. 2021.

[38] D. Kwon et al., "Design of secure handover authentication scheme for urban air mobility environments," *IEEE Access*, vol. 10, pp. 42529–42541, 2022.

[39] L. Meng, H. Xu, H. Xiong, X. Zhang, X. Zhou, and Z. Han, "An efficient certificateless authenticated key exchange protocol resistant to ephemeral key leakage attack for V2V communication in IoV," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11736–11747, Nov. 2021.

[40] 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.

[41] M. U. Javed, A. Jamal, E. H. Alkhammash, M. Hadjouni, S. A. Bahaj, and N. Javaid, "Secure message handling in vehicular energy networks using blockchain and artificially intelligent IPFS," *IEEE Access*, vol. 10, pp. 82063–82075, 2022.

[42] Q. Feng, D. He, S. Zeadally, and K. Liang, "BPAS: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4146–4155, Jun. 2020.

[43] 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," *Accident Anal. Prevention*, vol. 146, 2020, Art. no. 105711.

[44] T. Gao, X. Deng, N. Guo, and X. Wang, "An anonymous authentication scheme based on PMIPv6 for VANETs," *IEEE Access*, vol. 6, pp. 14686–14698, 2018.

[45] 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.

[46] 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.

[47] X. Zhang and X. Chen, "Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network," *IEEE Access*, vol. 7, pp. 58241–58254, 2019.

[48] D. Zheng, C. Jing, R. Guo, S. Gao, and L. Wang, "A traceable blockchain-based access authentication system with privacy preservation in VANETs," *IEEE Access*, vol. 7, pp. 117716–117726, 2019.

[49] A. Vangala, B. Bera, S. Saha, A. K. Das, N. Kumar, and Y. 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.

[50] P. Vijayakumar, M. Azees, S. A. Kozlov, and J. J. Rodrigues, "An anonymous batch authentication and key exchange protocols for 6 g enabled vanets," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1630–1638, Feb. 2022.

[51] P. Bagga, A. K. Das, M. Wazid, J. J. P. C. Rodrigues, K.-K. R. Choo, and Y. Park, "On the design of mutual authentication and key agreement protocol in internet of vehicles-enabled intelligent transportation system," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1736–1751, Feb. 2021.

[52] M. S. Kakkasageri and S. S. Manvi, "Multiagent driven dynamic clustering of vehicles in VANETs," *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1771–1780, 2012.

[53] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[54] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[55] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptography*, 2005, pp. 65–84.

[56] L. Kocarev and S. Lian, "Chaos-based cryptography: Theory, algorithms and applications," in *SCI Book Series*. Berlin, Germany: Springer, 2011.

[57] P. Bergamo, P. D,Arco, A. D. Santis, and L. Kocarev, "Security of public-key cryptosystems based on chebyshev polynomials," *IEEE Trans. Circuits Syst.*, vol. 52, no. 7, pp. 1382–1393, Jul. 2005.

[58] W. E. May, "Secure hash standard," 2015, FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U. S. Department of Commerce, 1995, Accessed: Jan. 2022. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

[59] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, 2001.

[60] H. Goumidi, Z. Aliouat, and M. Aliouat, "Efficient broadcast of alert messages in VANETs," in *Proc. IFIP Int. Conf. Comput. Intell. Appl.*, 2018, pp. 448–459.

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

[62] M. Abadi, B. Blanchet, and H. Comon-Lundh, "Models and proofs of protocol security: A progress report," in *Proc. 21st Int. Conf. Comput. Aided Verification*, 2009, pp. 35–49.

[63] M. Wazid, A. K. Das, R. Hussain, N. Kumar, and S. Roy, "BUAKA-CS: Blockchain-enabled user authentication and key agreement scheme for crowdsourcing system," *J. Syst. Archit.*, vol. 123, 2022, Art. no. 102370.

[64] C. Xu, X. Huang, M. Ma, and H. Bao, "An anonymous handover authentication scheme based on LTE-A for vehicular networks," *Wireless Commun. Mobile Comput.*, vol. 2018, 2018, Art. no. 6251219.

[65] "MIRACL cryptographic SDK: Multiprecision integer and rational arithmetic cryptographic library," 2020, Accessed: Jun. 2022. [Online]. Available: https://github.com/miracl/MIRACL

[66] "ns-3 network simulator," 2022, Accessed: Jun., 2022. [Online]. Available: https://www.nsnam.org/

[67] Intel Corporation, "Hyperledger sawtooth architecture guide," 2020, Accessed: Jan., 2023. [Online]. Available: https://sawtooth.hyperledger.org/docs/1.2/