Blockchain-assisted Lightweight Authenticated Key Agreement Security Framework for Smart Vehicles-enabled Intelligent Transportation System

Akhtar Badshah, Ghulam Abbas, *Senior Member, IEEE*, Muhammad Waqas, *Senior Member, IEEE*,
Fazal Muhammad, Ziaul Haq Abbas, Muhammad Bilal, *Senior Member, IEEE,* and
Houbing Song, *Fellow, IEEE*

***Abstract*—Intelligent Transportation Systems (ITS) supported by smart vehicles have revolutionized modern transportation, offering a wide range of applications and services, such as electronic toll collection, collision avoidance alarms, real-time parking management, and traffic planning. However, the open communication channels among various entities, including smart vehicles, roadside infrastructure, and fleet management systems, introduce security and privacy vulnerabilities. To address these concerns, we propose a novel security framework, named blockchain-assisted lightweight authenticated key agreement security framework for smart vehicles-enabled ITS (BASF-ITS), which ensures data protection both during transit and while stored on cloud servers. BASF-ITS employs a combination of efficient cryptographic primitives, including hash functions, XOR operator, ASCON, elliptic curve cryptography, and physical unclonable functions (PUF), to design authenticated key agreement schemes. The inclusion of PUF significantly enhances the system's resistance to physical attacks, preventing tampering attempts. To ensure data integrity when stored on the cloud, our framework incorporates blockchain technology. By leveraging the immutability and decentralization of the blockchain, BASF-ITS effectively safeguards data at rest, providing an additional layer of security. We rigorously analyze the security of BASF-ITS and demonstrate its strong resistance against potential security ass aults, making it a robust and reliable solution for smart vehicle-enabled ITS. In a comparative analysis with contemporary competing schemes, BASF-ITS emerges as a promising approach, offering superior functionality traits, enhanced security features, and reduced computation, communication, and storage costs. Furthermore, we present a practical implementation of BASF-ITS using blockchain technology, showcasing the computational time versus the "transactions per block" and the "number of mined blocks", confirming its efficiency and viability in real-world scenarios.**

A. Badshah is with the Department of Software Engineering, University of Malakand, Dir Lower 18800, Pakistan (e-mail: akhtarbadshah@uom.edu.pk).

G. Abbas is with the Telecommunication and Networking (TeleCoN) Research Center, Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan (e-mail: abbasg@giki.edu.pk).

M. Waqas is with the School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, United Kingdom, and also with the School of Engineering, Edith Cowan University, Perth, 6007 WA, Australia (e-mail: engr.waqas2079@gmail.com).

F. Muhammad is with the Department of Electrical Engineering, University of Engineering and Technology, Mardan 23200, Pakistan (e-mail: fazal.muhammad@uetmardan.edu.pk).

Z. H. Abbas is with the Faculty of Electrical Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan (e-mail: ziaul.h.abbas@giki.edu.pk).

M. Bilal is with the School of Computing and Communications, Lancaster University, Lancaster LA1 4WA, United Kingdom (e-mail: m.bilal@ieee.org).

H. Song is with the Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250 USA (e-mail: h.song@ieee.org; songh@umbc.edu).

***Note to Practitioners*—This article is motivated by designing an efficient, lightweight, and anonymous blockchain-enabled authenticated security framework that can fix the security and privacy concerns in insecure environments for ITS applications, such as automated road speed enforcement, collision avoidance alarm systems, and traffic planning and management, etc. Authenticated key agreement schemes are extensively used to secure communications in the ITS environment. However, the existing state-of-the-art schemes are not efficient in terms of performance, are not resilient against potential security attacks, and do not support anonymity, untraceability, and unlinkability. Therefore, we propose the authenticated security framework to secure communication among the participating entities in the ITS environment. It utilizes efficient cryptographic primitives, such as hash function, XOR-operator, ASCON, elliptic curve cryptography, and PUF. It is shown that the proposed framework can be deployed as a robust tool to address the ITS security problems efficiently. Moreover, the proposed framework is lightweight and efficient and can be easily deployed in various ITS applications and other resource-constrained environments. However, the participating entities, such as vehicles and roadside units, must be PUF-enabled to deploy the proposed framework.**

***Index Terms*—Internet of vehicles, intelligent transportation system, authentication, key agreement, blockchain, security.**

I. INTRODUCTION

In recent years, the Internet of Things (IoT) and Industry 4.0 have revolutionized various industries [1]–[6], and the concept of the Internet of Vehicles (IoV) has emerged as a crucial and integral component of intelligent transportation systems (ITSs). IoV is a distributed network that permits the use of data produced by smart vehicles and vehicular ad hoc networks (VANETs) [7]. An essential purpose of the IoV is to permit smart vehicles to communicate wirelessly in real-time with other smart vehicles, vehicle drivers, pedestrians, roadside units (RSUs), and network infrastructure to provide enhanced transportation applications and services, including but not limited to accident avoidance, traffic congestion reduction, route navigation, and infotainment [8]. Nevertheless, the conventional IoV adheres to the traditional centralized model, wherein a central server acts as a service provider for all the interconnected smart vehicles and RSUs. These devices communicate bi-directionally with the cloud. Unfortunately, this conventional IoV approach is plagued by a single point of vulnerability. The centralized server can also be compromised by numerous potential security attacks, such as data tempering and denial-of-service (DoS) attacks. Further, the transmitted data can be intercepted and tempered by adversaries [9], [10].

Blockchain technology has desirable traits of immutability, decentralization, integrity, transparency, auditability, anonymity, autonomy, and fault tolerance. The integration of blockchain technology with IoV represents the crucial missing element that has the potential to address various challenges in the smart transportation system. By incorporating blockchain, we can effectively tackle issues such as the heterogeneity of IoV systems, limitations in resources for end devices, network complexity, and security and privacy vulnerabilities [11]. Due to its decentralized and pliable data structure, blockchain technology does not suffer from a single point of vulnerability and is resilient to data-tempering attacks. Other characteristics of the blockchain technology include transparency, immutability, enhanced security, and preserving the integrity, privacy, and confidentiality of IoV applications [12]–[14].

Smart vehicles-enabled ITS assists and provides numerous services and applications. However, communicating entities, such as smart vehicles, RSUs, and cloud servers (CSs) communicate via wireless channels. An adversary will have the chance to intercept the transmitted messages as well as delete, modify, or insert the messages in such a communication environment. Moreover, the adversary can launch various potential security attacks in this environment, including man-in-the-middle (MitM), replay, impersonation, physical attacks, and ephemeral secret leakage (ESL) attacks, etc. Furthermore, untraceability and anonymity are crucial security traits in the ITS communication environment. The foremost defense against security attacks lies in implementing a proficient and trustworthy authenticated key agreement (AKA) scheme. Using this scheme, entities such as smart vehicles, RSUs, and CSs can achieve mutual authentication and establish secure session keys to facilitate their communication. In this context, the indispensable role of blockchain technology becomes evident, as it offers anonymity, decentralization, tamper-proof capabilities, and robust protection against various information security attacks. Hence, it is crucial to furnish a blockchain-enabled AKA scheme for smart vehicles-assisted ITS communication. Consequently, we introduce a novel security framework, called blockchain-assisted lightweight authenticated key agreement security framework for smart vehicles-enabled ITS (BASF-ITS). This framework incorporates AKA schemes to enable secure session key agreements between vehicle-to-vehicle (V2V) and vehicle-to-RSU (V2RSU) interactions, allowing IoV entities to transmit their data securely.

The paper's notable contributions are outlined as follows:

1) Our proposed framework, named BASF-ITS, addresses the critical issue of data security during transit by designing AKA schemes that incorporate a combination of efficient cryptographic primitives, including hash functions, XOR operator, ASCON, elliptic curve cryptography (ECC), and physical unclonable function (PUF). Through a comprehensive security analysis, we have demonstrated that our devised framework is resilient against potential security assaults. Notably, the incorporation of the PUF trait empowers smart vehicles and RSUs to thwart tampering from physical attacks, thus, significantly enhancing the overall system security.
2) To ensure integrity and protection of the data stored on cloud server, we have incorporated blockchain technology into our framework. By leveraging the inherent immutability and decentralized nature of the blockchain, our framework effectively safeguards data at rest from tampering attempts, providing an additional layer of security.
3) The comparative analysis of BASF-ITS with its contemporary competing schemes reveals that BASF-ITS provides more functionality traits, supports better security, and demands fewer computation, communication, and storage costs.
4) We have also executed the blockchain-based implementation of BASF-ITS to measure the computational time required for different numbers of transactions per block and various numbers of blocks mined in the blockchain.

The rest of this paper is structured as follows. The literature review on the existing relevant schemes is presented in Section II. The system model, adversary model, and essential preliminaries are given in Section III. We present our proposed BASF-ITS framework and the security analysis in Sections IV and V, respectively. In Section VI, we furnish a detailed implementation of the blockchain employed in BASF-ITS. A rigorous comparative analysis of our proposed BASF-ITS with the competing schemes is given in Section VII. Finally, we conclude this paper in Section VIII.

II. RELATED WORK

This section surveys the eminent and relevant access control, key management, and authentication schemes developed for the ITS and other pertinent environments. To this end, the authors in [15] offered a survey article spotlighting security requirements and potential security assaults in the IoVs environment. Further, the authors briefly discussed network and adversary models, the taxonomy of security protocols, tools for comparative analysis, numerous testbeds implementations, and various open issues and challenges associated with the security in IoVs. Mollah *et al.* [16] recently presented an extensive survey for blockchain-enabled applications in the IoV network. The authors explored several research areas in which blockchain is utilized in conjunction with IoVs to realize the vision of ITS and highlighted the benefits of blockchain-enabled IoVs applications, including security, immutability, automation, traceability, and decentralization.

Gupta *et al.* [17] proposed a quantum-defended blockchain-enabled data authentication scheme for IoVs. They utilized lattice cryptography to prevent quantum and data forging attacks and support data exchange, security, and credibility in addition to legitimate batch verification. Additionally, blockchain technology is used to protect the vehicles' public data in order to ensure their privacy swiftly.

Roy *et al.* [18] proposed a scheme for the ITS environment that utilized an extended chaotic map and access control mechanisms based on Chebyshev polynomials for communication between vehicles and roadside units. The comparative analysis demonstrated that employing these extended chaotic maps significantly enhanced the efficiency of the access control policy compared to baseline policies.

Xie *et al.* [19] proposed a protocol enabling secure vehicle-to-infrastructure (V2I), V2I handover, and V2V broadcasting authentications across diverse scenarios. This protocol incorporated security measures, such as PUF and bioinformation, to prevent attacks. Additionally, it included a dynamic anonymity strategy to prevent tracking and an identity recovery system for identifying malicious message senders.

Liu *et al.* [20] introduced the anonymous, traceable, and revocable credential system (ATRC), leveraging blockchain technology. ATRC, based on a versatile group signature framework, addresses the need for an anonymous yet traceable and revocable credentialing system. This innovative approach not only ensures user anonymity but also grants individuals greater control over managing their identities. The underlying group signature mechanism forms the backbone of ATRC, enabling users to navigate identity management securely within the system.

A novel AKA scheme was developed by Azees *et al.* for VANETs, as documented in their work [21]. This scheme effectively tackles the issue of revocation for malicious entities within the network. Notably, their proposed solution demonstrates computational efficiency in both certificate and signature verification processes. In addition, the authors of [22] devised a handover authentication scheme specifically designed to reduce the overhead associated with the re-authentication of vehicles. However, employing bilinear pairings operations makes the schemes in [21] and [22] computationally expensive.

The authors of [28] proposed a certificateless key agreement protocol for blockchain-enabled ITS. However, their proposed protocol does not support characteristics like untraceability and anonymity and also imposes significant communication and computational costs [23], [24]. Furthermore, a malicious adversary can utilize power analysis attacks to extract secret credentials stored in stolen/lost smart card or on-board unit (OBU) [25].

Karim *et al.* [26] introduced a blockchain-based solution with the goal of enhancing the security of IoV operating in a 5G environment. Their proposed scheme utilizes ECC for implementation. While their approach demonstrates potential, conducting a comprehensive evaluation of the methodology, underlying assumptions, and potential limitations is imperative to ensure the reliability and suitability of their proposed solution.

Xi *et al.* [27] presented an efficient anonymous authentication approach for the IoV using zero-knowledge proof and ECC. The main objective of their proposed scheme is to enhance user privacy and service efficiency within the IoV ecosystem by ensuring robust anonymity and authenticity during vehicle authentication. To address traceability concerns, the scheme incorporates a trusted authority for user traceability in case of violations. Additionally, Xi *et al.* included a fast reconnection procedure to minimize computation overhead, further optimizing the system's performance.

Vangala *et al.* [29] developed a certificate-based AKA scheme for blockchain-enabled ITS. Their proposed scheme ensures the secure transmission of accidental alerts among IoVs entities and facilitates the transfer of essential consensus-related information to the blockchain network. Nonetheless, it is important to note that the scheme lacks untraceability and anonymity features.

The author of [30] developed an efficient mutual authentication and key management scheme for the Internet of Drones (IoD) applications employing bilinear pairings, ECC, hash functions, and symmetric key encryption. However, the scheme is vulnerable to ESL attacks and does not support dynamic node addition, blockchain technology, and characteristics like untraceability and anonymity. Ali *et al.* [31] devised a user AKA scheme for the IoD environment utilizing a fuzzy extractor for biometric verification, a hash function, and symmetric encryption to enhance security. However, it is important to note that the scheme's vulnerability to chosen plaintext attacks raises concerns about its overall security. Additionally, the scheme does not incorporate support for blockchain technology, which could potentially enhance its security and reliability

Tanveer *et al.* [32] devised a user authentication scheme based on a hash function and an authenticated encryption with associative data (AEAD) scheme. Nevertheless, the scheme does not provide the feature of untraceability. Subsequently, the scheme proposed in [33] cannot withstand desynchronization and privileged insider attacks. The scheme proposed in [34] lacks the session key verification and anonymity features. The authors in [35] presented an AKA scheme based on ECC, hash function, and AEAD, which lacks the session key verification trait.

Wazid *et al.* [36] recently presented a secure framework for ITS using a public blockchain. The created framework guarantees secure communications between the communicating entities via AKA schemes. To ensure the integrity and confidentiality of the ITS, they use ElGamal-type signatures, certificates, and hashing algorithms in their proposed schemes. However, their proposed V2V AKA scheme is inaccurate due to the wrong computation of the certificates, which leads to the authenticity problem and ultimately fails to establish a secure session key between the vehicles.

III. SYSTEM MODEL AND PRELIMINARIES

In this section, we briefly introduce the system model of the proposed BASF-ITS framework. This section also discusses an adversary's capabilities and some essential preliminaries to design BASF-ITS.

A. System model

The system model for BASF-ITS is provided in Fig. 1, which consists of four entities: registration authority (RA), smart vehicles, RSUs, and CSs. There are various forms of communication, including V2V, V2RSU, and RSU2CS. Before inclusion in the network, registering each unique entity is the duty of RA. The RSUs are installed after loading the secret parameters in their memory. The essential credentials are also stored in the respective OBU of the smart vehicle and CS to use these secret credentials for further authentication and key agreement procedures. Each smart vehicle is connected with an RSU or another nearby smart vehicle via cellular networks or dedicated short-range communications.

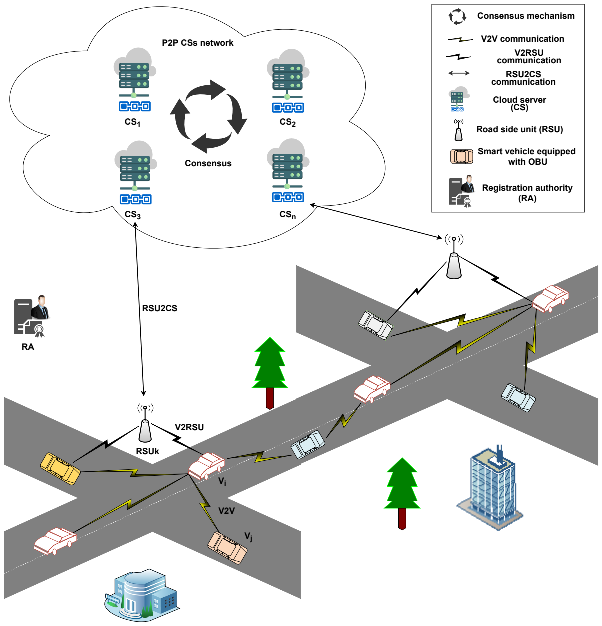

Fig. 1: System model for the proposed BASF-ITS framework.

Additionally, RSUs can connect via wired or wireless networks to the CS. Unfortunately, such kinds of communications are susceptible to adversaries and are exposed to numerous potential security attacks. The wireless channel's openness in vehicular networks naturally tempts attackers to undertake various assaults. Furthermore, blockchain technology accumulates the ITS environment's data across the peer-to-peer (P2P) CSs network (CSN). Blockchain technology offers protection from several possible attacks, including data disclosure and data modification assaults. This paper furnishes the details of the AKA process for the following two cases, V2V and V2RSU. After a successful AKA process, the communicating entities establish a secure secret session key for future secure communication.

B. Adversary model

For the devised BASF-ITS, we employ the widely utilized Dolev-Yao (DY) adversary model. According to the DY model, the communicated entities communicate across open channels, vulnerable to eavesdropping and other potential security attacks [37]. Smart vehicles and RSUs are examples of endpoint entities that are generally not trustworthy. Consequently, the transmitted messages may undergo modifications, drops, or delays. Additionally, the RA, responsible for entity registration, is considered a fully trusted network entity, while CSs are viewed as semi-trusted entities within the ITS environment. Furthermore, we adhere closely to the principles outlined in "Canetti and Krawczyk's (CK) adversary model," which possesses greater strength than the DY model and finds applications in the key establishment, access control, and authentication methods [38]. In the CK-adversary model, an adversary $\mathcal{A}$ is equipped with all the functionalities available in the DY model, alongside additional capabilities, including the ability to compromise secret credentials through session-hijacking attacks. Further, $\mathcal{A}$ can compromise the physical security of stolen/captured OBU of smart vehicles by launching power analysis attacks [39]. The extracted information can then be used by $\mathcal{A}$ to launch other potential security attacks, like unauthorized session key computation and impersonation attacks.

TABLE I: Notations of the proposed BASF-ITS

Symbol	Definition
$\mathcal{A}$	Adversary
CT, MAC	Ciphertext and authentication parameter generated by ASCON
$E_q(\alpha, \beta)$	Elliptic curve
P	Base point
$\oplus$, $\|$	XOR operation, Concatenation
RA	Registration authority
(s_x, Q_x)	A private-public key pair of an entity x
ID_x, PID_x	Real-identity and pseudo-identity of the entity x
$PUF(\cdot)$	Physical unclonable function
(C_x, R_x)	Challenge-response pair of the entity x
r_x	Random secret of an entity x utilized in the AKA phase
$h(\cdot)$	Collision-resistant hash function
TS_x	Timestamp picked by an entity x in the AKA phase
RT_x	Registration timestamp of an entity x
V_i, RSU_k, CS_l	ith smart vehicle, kth roadside unit, lth CS
VO_i, PW_{VO_i}	ith vehicle owner, owner's password
rn_x	Random nonce used in registration phase for entity x
ΔT	Maximum message delay

C. Preliminaries

The essential preliminaries employed for designing the proposed BASF-ITS are detailed below. Table I contains the notations utilized throughout this paper.

1) ASCON

ASCON stands as a renowned AEAD primitive, which ensures the preservation of data integrity, confidentiality, and authenticity, all accomplished without the use of a message authentication code. Distinguished by its inverse-free, single pass, and online symmetric block cipher characteristics [40], ASCON's encryption and decryption procedures can be summed up as follows.

$$(CT, MAC) = \mathcal{E}_K\{(AD, Nn), PT\},$$
$$(PT, MAC') = \mathcal{D}_K\{(AD, Nn), CT\},$$

where $AD, Nn, PT, CT, MAC/MAC'$, and K signify the associative data, nonce, plaintext, ciphertext, authentication parameters, and key, respectively. This article utilizes ASCON as the encryption/decryption primitive to design efficient and lightweight AKA schemes for the ITS environment.

2) Physical unclonable function

A PUF refers to a distinctive physical trait present in electronic devices, akin to a human's unique fingerprint. The most precise definition of a PUF is provided in [41], which characterizes it as "*an inherent and unclonable instance-specific feature of a physical object*." PUFs are based on two fundamental concepts introduced in [41]: intradistance and interdistance. These definitions establish the key characteristics of PUFs, including reproducibility, identifiability, uniqueness, physical unclonability, and unpredictability [41], [42]. Leveraging these distinctive properties, PUFs find extensive utility in identification, key generation, and authentication schemes.

Essentially, a PUF can be perceived as a function that relies on the complex physical structure of a circuit, wherein it maps a set of challenges to corresponding responses. The relationship between the challenge (C) and response (R) pair of the PUF can be represented as follows:

$$R = PUF(C).$$

IV. The Proposed Blockchain-assisted Framework

This section explains our devised BASF-ITS for communicating entities in the ITS environment, such as smart vehicles, RSUs, and CSs. After executing all steps of BASF-ITS, a secret session key is established for secure communication between a smart vehicle and the other neighboring smart vehicles, smart vehicle to the RSU, and RSU to the CS. The integration of blockchain constructs our system as more decentralized, reliable, and secure, which are the crucial necessities of an ITS. Our designed BASF-ITS comprises the trailing phases.

A. Setup phase

The RA is considered a fully trusted entity in the ITS environment. The responsibility of RA is to furnish offline tasks, including assigning an identity to each network entity and selecting cryptographic parameters and primitives. The security parameters of the RA for setting up the system are as follows.

To begin the setup process, the RA chooses a nonsingular elliptic curve denoted as $E_q(\alpha, \beta)$ over the Galois field $GF(q)$. This curve is defined by the equation $y^2 = x^3 + \alpha x + \beta \pmod{q}$, where α and β are constants taken from the set $Z_q = \{0, 1, 2, \cdots, q-1\}$. The condition $4\alpha^3 - 27\beta^2 \neq 0 \pmod{q}$ must be satisfied. Next, the RA selects a base point P belonging to the elliptic curve $E_q(\alpha, \beta)$, such that its order n_P is as large as q. In mathematical terms, this means that $n_P \cdot P = P + P + \cdots + P$ (where P is summed with itself n_P times) resulting in the "point at infinity" or the zero point denoted as $\mathcal{O}$. Additionally, the RA chooses a collision-resistant one-way cryptographic hash function denoted as $h(\cdot)$. Along with this hash function, a master secret key s_{RA} is randomly selected from the set $Z_q^* = \{1, 2, \cdots, q-1\}$, and the corresponding public key Q_{RA} is calculated as the scalar multiplication of the base point P with the secret key s_{RA}, i.e., $Q_{RA} = s_{RA} \cdot P$. The RA publishes $\{E_q(\alpha, \beta), P, h(\cdot), Q_{RA}\}$ and keeps s_{RA} confidential.

B. Registration phase

Before inclusion into the ITS environment and accessing the smart transportation services, individual network entities like vehicles, RSUs, and CSs must complete registration with the RA. The registration process is detailed below.

1) Vehicle registration phase

To gain access to the smart transportation application, the vehicle owner is required to register their vehicle with RA. The process of vehicle registration (VR) unfolds as follows:

VR-1: The registration process for a smart vehicle denoted as V_i, begins by sending a registration request to RA along with the vehicle's identity, represented as ID_{V_i}. Subsequently, RA selects a unique challenge parameter, denoted as C_{V_i}, and computes a pseudo-identity PID_{V_i} for V_i as $X_{V_i} = h(ID_{V_i} \parallel s_{RA} \parallel RT_{V_i})$ and $PID_{V_i} = X^1_{V_i} \oplus X^2_{V_i}$, where RT_{V_i} is the registration timestamp of V_i, and $X^1_{V_i}$ and $X^2_{V_i}$ are two equal chunks of 128 bits each as the size of X_{V_i} is 256 bits. Furthermore, RA generates a private key, denoted as s_{V_i}, specifically for V_i. Using this private key, RA calculates the corresponding public key, denoted as Q_{V_i}, as $Q_{V_i} = s_{V_i} \cdot P$. Further, RA store the parameters $\{s_{V_i}, Q_{V_i}, C_{V_i}, ID_{V_i}, PID_{V_i}\}$ in the OBU of V_i and publishes parameter Q_{V_i} publicly.

VR-2: After storing the credentials $\{s_{V_i}, Q_{V_i}, C_{V_i}, ID_{V_i}, PID_{V_i}\}$ in the OBU of V_i, the vehicle owner picks a password PW_{V_i}. Further, V_i generates a random nonce rn_{V_i} and calculates $R_{V_i} = PUF(C_{V_i})$, $A_{V_i} = h(ID_{V_i} \parallel PW_{V_i} \parallel R_{V_i})$, and $K_{V_i} = A^1_{V_i} \oplus A^2_{V_i}$. Moreover, V_i using ASCON encryption computes $(CT_{V_i}, MAC_{V_i}) = \mathcal{E}_{K_{V_i}}\{(AD_1, Nn_1), PT_{V_i}\}$, where $AD_1 = rn_{V_i}$, $Nn_1 = A^2_{V_i}$ and $PT_{V_i} = \{PID_{V_i}, s_{V_i}\}$.

VR-3: Finally, V_i keeps the following secret credentials $\{CT_{V_i}, MAC_{V_i}, rn_{V_i}, C_{V_i}, PUF(\cdot), Q_{V_i}\}$ within its onboard unit OBU_{V_i} prior to deployment.

It is crucial to highlight that the vehicle owner must remember PW_{V_i} to access the smart transportation service.

Remark 1. *It is important to note that the majority of AEAD schemes use 128-bit-sized AD, nonce, and key. In this context, K_{V_i} is obtained by XORing two equal 128-bit chunks, $A^1_{V_i}$ and $A^2_{V_i}$, from the 256-bit-sized A_{V_i}. As a result, K_{V_i} has a size of 128 bits. For our study, we utilize the ASCON algorithm, and we will follow the aforementioned process to derive all the required parameters compatible with the AEAD encryption technique.*

2) RSU registration phase

The RA also carries out the registration procedure of an RSU RSU_k utilizing the trailing steps:

RR-1: The process of adding an RSU_k begins with an initiation of a registration procedure by RA. During this procedure, RA selects a distinctive challenge parameter, denoted as C_{RSU_k}, and securely transmits it to RSU_k through a secure channel. Subsequently, RSU_k computes the response parameter, represented as $R_{RSU_k} = PUF(C_{RSU_k})$, and securely sends it back to RA via a secure channel.

RR-2: For RSU_k, RA first chooses a private key s_{RSU_k} and then computes the corresponding public key Q_{RSU_k} as $Q_{RSU_k} = s_{RSU_k} \cdot P$. Furthermore, RA also chooses a unique real identity ID_{RSU_k} and determines a pseudo-identity PID_{RSU_k} of RSU_k by computing $X_{RSU_k} = h(ID_{RSU_k} \parallel s_{RA} \parallel RT_{RSU_k})$ and $PID_{RSU_k} = X^1_{RSU_k} \oplus X^2_{RSU_k}$, where RT_{RSU_k} is the registration timestamp of RSU_k.

RR-3: Further, RA generates a random nonce rn_{RSU_k} and calculates $A_{RSU_k} = h(ID_{RSU_k} \parallel R_{RSU_k})$, and $K_{RSU_k} = A^1_{RSU_k} \oplus A^2_{RSU_k}$. Moreover, RSU_k using ASCON encryption computes $(CT_{RSU_k}, MAC_{RSU_k}) = \mathcal{E}_{K_{RSU_k}}\{(AD_1, Nn_2), PT_{RSU_k}\}$, where $AD_1 = rn_{RSU_k}$, $Nn_2 = A^2_{RSU_k}$, and $PT_{RSU_k} = \{PID_{RSU_k}, s_{RSU_k}\}$.

RR-4: Finally, the RA stores the following secret credentials $\{CT_{RSU_k}, MAC_{RSU_k}, rn_{RSU_k}, C_{RSU_k}, PUF(\cdot), Q_{RSU_k}\}$ in RSU_k's memory before its deployment and publishes the parameter Q_{RSU_k}.

3) CS registration

Prior to deploying CS CS_l, the RA carries out the CS registration (CR) process using the following steps.

CR-1: To begin the CR process for CS_l, RA selects a unique real identity ID_{CS_l}. Subsequently, RA determines a pseudo-identity PID_{CS_l} for CS_l by performing the computations $X_{CS_l} = h(ID_{CS_l} \parallel s_{RA} \parallel RT_{CS_l})$ and $PID_{CS_l} = X^1_{CS_l} \oplus X^2_{CS_l}$, where RT_{CS_l} is the registration timestamp of CS_l. Furthermore, RA chooses a private key s_{CS_l} and computes the corresponding public key as $Q_{CS_l} = s_{CS_l} \cdot P$.

CR-2: RA transmits the parameters $\{ID_{CS_l}, PID_{CS_l}, (s_{CS_l}, Q_{CS_l})\}$ to CS_l via a secure private channel utilizing a shared secret key K_{RA,CS_l} between them. Furthermore, via a secure channel, RA also sends the registration details of the vehicles and RSUs in a specific traffic zone to the relevant cloud server CS_l.

CR-3: Finally, after acquiring the registration credentials from RA, CS_l keeps the secret parameter $\{ID_{CS_l}, PID_{CS_l}, (s_{CS_l}, Q_{CS_l}), h(\cdot), P, E_q(\alpha, \beta)\}$ in its secure database and the parameter Q_{CS_l} is published.

Remark 2. *It is noteworthy that CS_l stores all its secret credentials in the secure region of its memory in order to protect it from stolen verifier attacks and other possible potential assaults.*

C. Vehicle user login phase

Upon successfully registering V_i, the vehicle user is required to perform a login process to access the smart transportation services running on V_i. The login procedure is outlined below.

UL-1: Vehicle user enters his/her password, denoted as $PW^l_{V_i}$, into smart application running on V_i.

UL-2: Smart application retrieves C_{V_i} from OBU_{V_i} and calculate the challenge parameter $R_{V_i} = PUF(C_{V_i})$. Subsequently, it calculates $A_{V_i} = h(ID_{V_i} \parallel PW^l_{V_i} \parallel R_{V_i})$, and $K_{V_i} = A^1_{V_i} \oplus A^2_{V_i}$, $AD^l = rn_{V_i}$, $Nn^l = A^2_{V_i}$. In addition, by using ASCON decryption function, the plaintext PT_{V_i}, which is equal to $PT_{V_i} = \{PID_{V_i}, s_{V_i}\}$, and authentication parameter $MAC^l_{V_i}$ are retrieved as $(PT_{V_i}, MAC^l_{V_i}) = \mathscr{D}_{K_{V_i}}\{(AD^l, Nn^l), CT_{V_i}\}$. Further, V_i checks if $MAC^l_{V_i} \overset{?}{=} MAC_{V_i}$ holds. If the verification is successful, it indicates that the vehicle user has been logged into V_i successfully.

D. Authenticated key agreement phase

This phase presents the proposed AKA schemes for the two different scenarios, i.e., between a vehicle V_i and an associated cluster head (CH) V_j and between a vehicle V_j, and its related RSU_k.

1) AKA between Vehicle V_i and Vehicle V_j

The steps outlined below must be accomplished in order to complete this task.

V2V-1: As the initiator vehicle, V_i generates a random nonce $r_{V_i} \in Z^*_q$ and then selects the current timestamp TS_{V_i}. Next, V_i computes $RB_{V_i} = r_{V_i} \cdot P$, $RSC_{V_iV_j} = r_{V_i} \cdot Q_{V_j}$, and $B_{V_i} = h(RSC_{V_iV_j} \parallel TS_{V_i})$. Furthermore, the secret encryption key K_1 is computed as $B^1_{V_i} \oplus B^2_{V_i}$, where $B^1_{V_i}$ and $B^2_{V_i}$ are derived from B_{V_i}. Moreover, by using ASCON encryption function, V_i computes $(CT_1, MAC_1) = \mathscr{E}_{K_1}\{(B^2_{V_i}, B^2_{V_i}), (PID_{V_i} \parallel Q_{V_i})\}$. Next, V_i forwards the message $msg_{VV_1} : \{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ to V_j via open channel.

V2V-2: V_j being the responder vehicle, after obtaining the message $msg_{VV_1} : \{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ at time TS'_{V_i}, first verifies the freshness of msg_{VV_1} by checking the condition $|TS_{V_i} - TS'_{V_i}| < \Delta T$. If the condition is satisfied, V_j proceeds to compute $RSC_{V_jV_i} = s_{V_j} \cdot RB_{V_i}$, $B_{V_j} = h(RSC_{V_jV_i} \parallel TS_{V_i})$, and $K'_1 = B^1_{V_j} \oplus B^2_{V_j}$. In addition, by using ASCON decryption function, the plaintext $\{PID_{V_i} \parallel Q_{V_i}\}$ and authentication parameter MAC_2 is retrieved as $(\{PID_{V_i} \parallel Q_{V_i}\}, MAC_2) = \mathscr{D}_{K'_1}\{(B^2_{V_j}, B^2_{V_j}), CT_1\}$. Next V_j checks if $MAC_2 \overset{?}{=} MAC_1$ holds. If so, then it generates TS_{V_j} and r_{V_j}. Next V_j computes $SC_{V_jV_i} = s_{V_j} \cdot Q_{V_i}$, $B1_{V_j} = h(SC_{V_jV_i} \parallel TS_{V_j})$, $K_2 = B1^1_{V_j} \oplus B1^2_{V_j}$, $SK_{V_jV_i} = h(SC_{V_jV_i} \parallel PID_{V_i} \parallel PID_{V_j} \parallel RB_{V_i} \parallel r_{V_j} \parallel TS_{V_i} \parallel TS_{V_j})$, $D_{V_j} = h(SK_{V_jV_i} \parallel TS_{V_i} \parallel TS_{V_j})$, and $SKV_{V_j} = D^1_{V_j} \oplus D^2_{V_j}$, where $SK_{V_jV_i}$ and SKV_{V_j} are the session key and session key verifier, respectively. Moreover, by using the ASCON encryption function, V_j computes $(CT_2, MAC_3) = \mathscr{E}_{K_2}\{(B1^2_{V_j}, B1^2_{V_j}), (PID_{V_j} \parallel SKV_{V_j} \parallel r_{V_j})\}$ and then sends the response message $msg_{VV_2} : \{CT_2, MAC_3, TS_{V_j}\}$ to V_i via open channel.

V2V-3: Upon receiving msg_{VV_2} from V_j, V_i verifies the freshness of the message msg_{VV_2} by checking the condition $|TS_{V_j} - TS'_{V_j}| < \Delta T$. If the condition is satisfied, V_i proceeds to compute $SC_{V_iV_j} = s_{V_i} \cdot Q_{V_j}$, $B1_{V_i} = h(SC_{V_iV_j} \parallel TS_{V_j})$, $K'_2 = B1^1_{V_i} \oplus B1^2_{V_i}$, and $((PID_{V_j} \parallel SKV_{V_j} \parallel r_{V_j}), MAC_4) = \mathscr{D}_{K_2}\{(B^2_{V_i}, B^2_{V_i}), CT_2\}$. Next, V_i checks the condition $MAC_3 \overset{?}{=} MAC_4$. If it holds, then V_i computes $SK_{V_iV_j} = h(SC_{V_iV_j} \parallel PID_{V_i} \parallel PID_{V_j} \parallel RB_{V_i} \parallel r_{V_j} \parallel TS_{V_i} \parallel TS_{V_j})$, $D_{V_i} = h(SK_{V_iV_j} \parallel TS_{V_i} \parallel TS_{V_j})$, and $SKV_{V_i} = D^1_{V_i} \oplus D^2_{V_i}$. Further, V_i checks $SKV_{V_j} \overset{?}{=} SKV_{V_i}$. If it holds, V_i stores $SK_{V_iV_j} (= SK_{V_jV_i})$ as session key.

Fig. 2 presents a summary of the diverse messages exchanged throughout the AKA phase between two neighboring vehicles.

2) AKA between Vehicle V_i and RSU_k

The following steps outline the AKA procedure between the CH (e.g., V_i) and RSU RSU_k.

V2R-1: V_i starts the AKA process by picking a random secret $r_{V_i} \in Z^*_q$ and timestamp TS_{V_i}. Next, it calculates $RB_{V_i} = r_{V_i} \cdot P$, $RSC_{V_iRSU_k} = r_{V_i} \cdot Q_{RSU_k}$, $B_{V_i} = h(RSC_{V_iRSU_k} \parallel TS_{V_i})$, and $K_1 = B^1_{V_i} \oplus B^2_{V_i}$. Moreover, by using ASCON encryption function, V_i computes $(CT_1, MAC_1) = \mathscr{E}_{K_1}\{(B^2_{V_i}, B^2_{V_i}), (PID_{V_i} \parallel Q_{V_i})\}$. After the computations, V_i constructs the message $msg_{VR_1} : \{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ and sends it to RSU_k via an open channel.

V2R-2: Upon the arrival of msg_{VR_1} at time TS'_{V_i}, RSU_k checks the freshness by verifying the condition $|TS_{V_i} - TS'_{V_i}| < \Delta T$? If so, RSU_k retrieves the parameters C_{RSU_k} and rn_{RSU_k}, and calculates $R_{RSU_k} = PUF(C_{RSU_k})$, $A_{RSU_k} = h(ID_{RSU_k} \parallel R_{RSU_k})$, and $K_{RSU_k} = A^1_{RSU_k} \oplus A^2_{RSU_k}$. In addition, by using ASCON decryption function, the plaintext $PT_{RSU_k} = \{PID_{RSU_k}, s_{RSU_k}\}$ and authentication parameter $MAC^l_{RSU_k}$ is retrieved as $(PT_{RSU_k}, MAC^l_{RSU_k}) = \mathcal{D}_{K_{RSU_k}}\{(rn_{RSU_k}, A^2_{RSU_k}), CT_{RSU_k}\}$. Next, RSU_k checks if $MAC^l_{RSU_k} \overset{?}{=} MAC_{RSU_k}$ holds. If so, RSU_k further computes $RSC_{RSU_kV_i} = s_{RSU_k} \cdot RB_{V_i}$, $B_{RSU_k} = h(RSC_{RSU_kV_i} \parallel TS_{V_i})$, and $K'_1 = B^1_{RSU_k} \oplus B^2_{RSU_k}$. Moreover, by using ASCON decryption function, the plaintext $\{PID_{V_i} \parallel Q_{V_i}\}$ and authentication parameter MAC_2 is computed as $(\{PID_{V_i} \parallel Q_{V_i}\}, MAC_2) = \mathcal{D}_{K'_1}\{(B^2_{RSU_k}, B^2_{RSU_k}), CT_1\}$. Next, RSU_k checks if $MAC_2 \overset{?}{=} MAC_1$ holds. If so, it generates the current timestamp TS_{RSU_k} and a random nonce r_{RSU_k}. Further, RSU_k computes $SC_{RSU_kV_i} = s_{RSU_k} \cdot Q_{V_i}$, $B1_{RSU_k} = h(SC_{RSU_kV_i} \parallel TS_{RSU_k})$, $K_2 = B1^1_{RSU_k} \oplus B1^2_{RSU_k}$, $SK_{RSU_kV_i} = h(SC_{RSU_kV_i} \parallel PID_{V_i} \parallel PID_{RSU_k} \parallel RB_{V_i} \parallel r_{RSU_k} \parallel TS_{V_i} \parallel TS_{RSU_k})$, $D_{RSU_k} = h(SK_{RSU_kV_i} \parallel TS_{V_i} \parallel TS_{RSU_k})$, and $SKV_{RSU_k} = D^1_{RSU_k} \oplus D^2_{RSU_k}$, where $SK_{RSU_kV_i}$ and SKV_{RSU_k} are the session key and session key verifier, respectively. Moreover, by using the ASCON encryption function, RSU_k computes $(CT_2, MAC_3) = \mathcal{E}_{K_2}\{(B1^2_{RSU_k}, B1^2_{RSU_k}), (PID_{RSU_k} \parallel SKV_{RSU_k} \parallel r_{RSU_k})\}$, constructs the response message $msg_{VR_2} : \{CT_2, MAC_3, TS_{RSU_k}\}$, and then sends it to V_i via open channel.

V2R-3: After obtaining msg_{VR_2} from RSU_k, V_i checks the freshness of the message msg_{VR_2} by checking the condition $|TS_{RSU_k} - TS'_{RSU_k}| < \Delta T$? If it holds, then V_i computes $SC_{V_iRSU_k} = s_{V_i} \cdot Q_{RSU_k}$, $B1_{V_i} = h(SC_{V_iRSU_k} \parallel TS_{RSU_k})$, $K'_2 = B1^1_{V_i} \oplus B1^2_{V_i}$, and $((PID_{RSU_k} \parallel SKV_{RSU_k} \parallel r_{RSU_k}), MAC_4) = \mathcal{D}_{K_2}\{(B^2_{V_i}, B^2_{V_i}), CT_2\}$. Next, V_i checks the condition $MAC_3 \overset{?}{=} MAC_4$. If it holds, then V_i computes $SK_{V_iRSU_k} = h(SC_{V_iRSU_k} \parallel PID_{V_i} \parallel PID_{RSU_k} \parallel RB_{V_i} \parallel$

Vehicle V_i	**Vehicle** V_j
Known parameters:$\{CT_{V_i}, MAC_{V_i}, rn_{V_i}, C_{V_i}, PUF(\cdot), Q_{V_i}\}$ Input: $ID_{V_i}, PW^l_{V_i}$	Known parameters:$\{CT_{V_j}, MAC_{V_j}, rn_{V_j}, C_{V_j}, PUF(\cdot), Q_{V_j}\}$ Input: $ID_{V_j}, PW^l_{V_j}$
Retrieve: C_{V_i}, rn_{V_i}; Compute: $R_{V_i} = PUF(C_{V_i})$, $A_{V_i} = h(ID_{V_i} \parallel PW^l_{V_i} \parallel R_{V_i})$, $K_{V_i} = A^1_{V_i} \oplus A^2_{V_i}$, $AD^l = rn_{V_i}$, $Nn^l = A^2_{V_i}$, $(PT_{V_i}, MAC^l_{V_i}) = \mathcal{D}_{K_{V_i}}\{(AD^l, Nn^l), CT_{V_i}\}$; Check if $MAC^l_{V_i} \overset{?}{=} MAC_{V_i}$ holds: Vehicle driver is successfully login. $PT_{V_i} = \{PID_{V_i}, s_{V_i}\}$;	Retrieve: C_{V_j}, rn_{V_j}; Compute: $R_{V_j} = PUF(C_{V_j})$, $A_{V_j} = h(ID_{V_j} \parallel PW^l_{V_j} \parallel R_{V_j})$, $K_{V_j} = A^1_{V_j} \oplus A^2_{V_j}$, $AD^l = rn_{V_j}$, $Nn^l = A^2_{V_j}$, $(PT_{V_j}, MAC^l_{V_j}) = \mathcal{D}_{K_{V_j}}\{(AD^l, Nn^l), CT_{V_j}\}$; Check if $MAC^l_{V_j} \overset{?}{=} MAC_{V_j}$ holds: Vehicle driver is successfully login. $PT_{V_j} = \{PID_{V_j}, s_{V_j}\}$;
Mutual authentication and key agreement scheme	
Pick: random nonce r_{V_i}; Select: current timestamp TS_{V_i}; Calculate: $RB_{V_i} = r_{V_i} \cdot P$, $RSC_{V_iV_j} = r_{V_i} \cdot Q_{V_j}$, $B_{V_i} = h(RSC_{V_iV_j} \parallel TS_{V_i})$, $K_1 = B^1_{V_i} \oplus B^2_{V_i}$, $(CT_1, MAC_1) = \mathcal{E}_{K_1}\{(B^2_{V_i}, B^2_{V_i}), (PID_{V_i} \parallel Q_{V_i})\}$ $\xrightarrow[(V_i \to V_j)]{msg_{VV_1}:\{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}}$.	
	Check if $\lvert TS_{V_i} - TS'_{V_i} \rvert < \Delta T$? Compute: $RSC_{V_jV_i} = s_{V_j} \cdot RB_{V_i}$, $B_{V_j} = h(RSC_{V_jV_i} \parallel TS_{V_i})$, $K'_1 = B^1_{V_j} \oplus B^2_{V_j}$, $(\{PID_{V_i} \parallel Q_{V_i}\}, MAC_2) = \mathcal{D}_{K'_1}\{(B^2_{V_j}, B^2_{V_j}), CT_1\}$ Check if $MAC_2 \overset{?}{=} MAC_1$ holds; If so, generate: TS_{V_j} and R_b; Compute: $SC_{V_jV_i} = s_{V_j} \cdot Q_{V_i}$, $B1_{V_j} = h(SC_{V_jV_i} \parallel TS_{V_j})$, $K_2 = B1^1_{V_j} \oplus B1^2_{V_j}$, $SK_{V_jV_i} = h(SC_{V_jV_i} \parallel PID_{V_i} \parallel PID_{V_j} \parallel RB_{V_i} \parallel R_b \parallel TS_{V_i} \parallel TS_{V_j})$, $D_{V_j} = h(SK_{V_jV_i} \parallel TS_{V_i} \parallel TS_{V_j})$, $SKV_{V_j} = D^1_{V_j} \oplus D^2_{V_j}$, $(CT_2, MAC_3) = \mathcal{E}_{K_2}\{(B1^2_{V_j}, B1^2_{V_j}), (PID_{V_j} \parallel SKV_{V_j} \parallel R_b)\}$ $\xleftarrow[(V_j \to V_i)]{msg_{VV_2}:\{CT_2, MAC_3, TS_{V_j}\}}$.
Check if $\lvert TS_{V_j} - TS'_{V_j} \rvert < \Delta T$? If so, Compute: $SC_{V_iV_j} = s_{V_i} \cdot Q_{V_j}$, $B1_{V_i} = h(SC_{V_iV_j} \parallel TS_{V_j})$, $K'_2 = B1^1_{V_i} \oplus B1^2_{V_i}$, $((PID_{V_j} \parallel SKV_{V_j} \parallel R_b), MAC_4) = \mathcal{D}_{K_2}\{(B^2_{V_i}, B^2_{V_i}), CT_2\}$, Check if $MAC_3 \overset{?}{=} MAC_4$ holds; compute $SK_{V_iV_j} = h(SC_{V_iV_j} \parallel PID_{V_i} \parallel PID_{V_j} \parallel RB_{V_i} \parallel R_b \parallel TS_{V_i} \parallel TS_{V_j})$, $D_{V_i} = h(SK_{V_iV_j} \parallel TS_{V_i} \parallel TS_{V_j})$, $SKV_{V_i} = D^1_{V_i} \oplus D^2_{V_i}$, Check if $SKV_{V_j} \overset{?}{=} SKV_{V_i}$ hold; Store $SK_{V_iV_j} (= SK_{V_jV_i})$ as SK.	
$SK_{V_iV_j} (= SK_{V_jV_i}) = h((s_{V_i} \cdot s_{V_j} \cdot P) \parallel PID_{V_i} \parallel PID_{V_j} \parallel (r_{V_i} \cdot P) \parallel r_{V_j} \parallel TS_{V_i} \parallel TS_{V_j})$	

Fig. 2: Summary of the AKA phase between two neighboring vehicles V_i and V_j.

$r_{RSU_k} \parallel TS_{V_i} \parallel TS_{RSU_k})$, $D_{V_i} = h(SK_{V_iRSU_k} \parallel TS_{V_i} \parallel TS_{RSU_k})$, and $SKV_{V_i} = D^1_{V_i} \oplus D^2_{V_i}$. Further, V_i checks $SKV_{RSU_k} \stackrel{?}{=} SKV_{V_i}$. If it holds, V_i stores $SK_{V_iRSU_k}(= SK_{RSU_kV_i})$ as session key.

Fig. 3 presents a summary of the diverse messages exchanged throughout the AKA phase between the CH (V_i) and RSU RSU_j.

Remark 3. *It is worth mentioning that communicating entities, such as RSU_k and CS_l, are resource-rich devices deployed in the ITS environment. As a result, they can utilize their "ECC-based private-public key pair" for secure communication.*

E. Dynamic node addition phase

In order to incorporate a new vehicle V^n_i into the ITS environment, RA carries out the following essential steps.

DNAP1: To register a new vehicle, say V^n_i, the registration process begins by sending a request to the RA, along with the identity $ID_{V^n_i}$ of V^n_i. Subsequently, RA chooses a unique challenge parameter $C_{V^n_i}$ and computes a pseudo-identity $PID_{V^n_i}$ for V^n_i by computing $X_{V^n_i} = h(ID_{V^n_i} \parallel s_{RA} \parallel RT_{V^n_i})$ and $PID_{V^n_i} = X^1_{V^n_i} \oplus X^2_{V^n_i}$, where $RT_{V^n_i}$ is the registration timestamp of V^n_i. Furthermore, RA generates a private key, $s_{V^n_i}$, for the vehicle V^n_i. Subsequently, the corresponding public key, $Q_{V^n_i}$, is computed as $Q_{V^n_i} = s_{V^n_i} \cdot P$. Further, RA store the parameters $\{s_{V^n_i}, Q_{V^n_i}, C_{V^n_i}, ID_{V^n_i}, PID_{V^n_i}\}$ in the OBU of V^n_i and publishes the parameter $Q_{V^n_i}$ publicly.

DNAP2: Next, the vehicle owner picks a password $PW_{V^n_i}$. Further, V^n_i generates a random nonce $rn_{V^n_i}$ and calculates $R_{V^n_i} = PUF(C_{V^n_i})$, $A_{V^n_i} = h(ID_{V^n_i} \parallel PW_{V^n_i} \parallel R_{V^n_i})$, and $K_{V^n_i} = A^1_{V^n_i} \oplus A^2_{V^n_i}$. Moreover, V^n_i using ASCON encryption computes $(CT_{V^n_i}, MAC_{V^n_i}) = \mathcal{E}_{K_{V^n_i}}\{(AD_1, Nn_1), PT_{V^n_i}\}$, where $AD_1 = rn_{V^n_i}$, $Nn_1 = A^2_{V^n_i}$, and $PT_{V^n_i} = \{PID_{V^n_i}, s_{V^n_i}\}$.

DNAP3: Finally, the vehicle V^n_i maintains the following secret credentials in its on-board unit ($OBU_{V^n_i}$) before deployment: $\{CT_{V^n_i}, MAC_{V^n_i}, rn_{V^n_i}, C_{V^n_i}, PUF(\cdot), Q_{V^n_i}\}$.

Likewise, a new RSU RSU^n_k and CS CS^n_l can be registered in the ITS environment through the RA, as outlined in Subsection IV-B2 (RSU registration) and Subsection IV-B3 (CS registration), respectively, prior to their deployment.

Vehicle V_i	**Roadside Unit** RSU_k
$\{CT_{V_i}, MAC_{V_i}, rn_{V_i}, C_{V_i}, PUF(\cdot), Q_{V_i}\}$	$\{CT_{RSU_k}, MAC_{RSU_k}, rn_{RSU_k}, ID_{RSU_k}, C_{RSU_k}, PUF(\cdot), Q_{RSU_k}\}$
Pick: random nonce r_{V_i}; Select: current timestamp TS_{V_i}; Calculate: $RB_{V_i} = r_{V_i} \cdot P$, $RSC_{V_iRSU_k} = r_{V_i} \cdot Q_{RSU_k}$, $B_{V_i} = h(RSC_{V_iRSU_k} \parallel TS_{V_i})$, $K_1 = B^1_{V_i} \oplus B^2_{V_i}$, $(CT_1, MAC_1) = \mathcal{E}_{K_1}\{(B^2_{V_i}, B^2_{V_i}), (PID_{V_i} \parallel Q_{V_i})\}$ $\xrightarrow[(V_i \rightarrow RSU_k)]{msg_{VR_1}:\{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}}$.	Check if $\lvert TS_{V_i} - TS'_{V_i}\rvert < \Delta T$? Retrieve: C_{RSU_k}, rn_{RSU_k}; Compute: $R_{RSU_k} = PUF(C_{RSU_k})$, $A_{RSU_k} = h(ID_{RSU_k} \parallel R_{RSU_k})$, $K_{RSU_k} = A^1_{RSU_k} \oplus A^2_{RSU_k}$, $AD^l = rn_{RSU_k}$, $Nn^l = A^2_{RSU_k}$, $(PT_{RSU_k}, MAC^l_{RSU_k}) = \mathcal{D}_{K_{RSU_k}}\{(AD^l, Nn^l), CT_{RSU_k}\}$; Check if $MAC^l_{RSU_k} \stackrel{?}{=} MAC_{RSU_k}$ holds; if so $PT_{RSU_k} = \{PID_{RSU_k}, s_{RSU_k}\}$; Compute: $RSC_{RSU_kV_i} = s_{RSU_k} \cdot RB_{V_i}$, $B_{RSU_k} = h(RSC_{RSU_kV_i} \parallel TS_{V_i})$, $K'_1 = B^1_{RSU_k} \oplus B^2_{RSU_k}$, $(\{PID_{V_i} \parallel Q_{V_i}\}, MAC_2) = \mathcal{D}_{K'_1}\{(B^2_{RSU_k}, B^2_{RSU_k}), CT_1\}$ Check if $MAC_2 \stackrel{?}{=} MAC_1$ holds; If so, generate: TS_{RSU_k} and r_{RSU_k}; Compute: $SC_{RSU_kV_i} = s_{RSU_k} \cdot Q_{V_i}$, $B1_{RSU_k} = h(SC_{RSU_kV_i} \parallel TS_{RSU_k})$, $K_2 = B1^1_{RSU_k} \oplus B1^2_{RSU_k}$, $SK_{RSU_kV_i} = h(SC_{RSU_kV_i} \parallel PID_{V_i} \parallel PID_{RSU_k} \parallel RB_{V_i} \parallel r_{RSU_k} \parallel TS_{V_i} \parallel TS_{RSU_k})$, $D_{RSU_k} = h(SK_{RSU_kV_i} \parallel TS_{V_i} \parallel TS_{RSU_k})$, $SKV_{RSU_k} = D^1_{RSU_k} \oplus D^2_{RSU_k}$, $(CT_2, MAC_3) = \mathcal{E}_{K_2}\{(B1^2_{RSU_k}, B1^2_{RSU_k}), (PID_{RSU_k} \parallel SKV_{RSU_k} \parallel r_{RSU_k})\}$ $\xleftarrow[(RSU_k \rightarrow V_i)]{msg_{VR_2}:\{CT_2, MAC_3, TS_{RSU_k}\}}$.
Check if $\lvert TS_{RSU_k} - TS'_{RSU_k}\rvert < \Delta T$? If so, Compute: $SC_{V_iRSU_k} = s_{V_i} \cdot Q_{RSU_k}$, $B1_{V_i} = h(SC_{V_iRSU_k} \parallel TS_{RSU_k})$, $K'_2 = B1^1_{V_i} \oplus B1^2_{V_i}$, $((PID_{RSU_k} \parallel SKV_{RSU_k} \parallel r_{RSU_k}), MAC_4) = \mathcal{D}_{K_2}\{(B^2_{V_i}, B^2_{V_i}), CT_2\}$, Check if $MAC_3 \stackrel{?}{=} MAC_4$ holds; compute $SK_{V_iRSU_k} = h(SC_{V_iRSU_k} \parallel PID_{V_i} \parallel PID_{RSU_k} \parallel RB_{V_i} \parallel r_{RSU_k} \parallel TS_{V_i} \parallel TS_{RSU_k})$, $D_{V_i} = h(SK_{V_iRSU_k} \parallel TS_{V_i} \parallel TS_{RSU_k})$, $SKV_{V_i} = D^1_{V_i} \oplus D^2_{V_i}$, Check if $SKV_{RSU_k} \stackrel{?}{=} SKV_{V_i}$ hold; Store $SK_{V_iRSU_k}(= SK_{RSU_kV_i})$ as SK.	
$SK_{V_i,RSU_k}(= SK_{RSU_k,V_i}) = h((s_{V_i} \cdot s_{RSU_k} \cdot P) \parallel PID_{V_i} \parallel PID_{RSU_k} \parallel (r_{V_i} \cdot P) \parallel r_{RSU_k} \parallel TS_{V_i} \parallel TS_{RSU_k})$.	

Fig. 3: Summary of the AKA phase between vehicle V_i and RSU RSU_k.

F. Password reset phase

Vehicle owner VO_i can change or reset his/her password for security reasons. To do this, VO_i must execute the smart application running on his/her smart vehicle (for instance, V_i). The following essential steps are then performed as part of the password change/reset procedure.

PR-1: The smart application displays a prompt requesting VO_i to input their password. Subsequently, VO_i enters the password $PW^l_{V_i}$.

PR-2: Smart application retrieves C_{V_i} from OBU_{V_i} and calculates the response parameter as $R_{V_i} = PUF(C_{V_i})$. Additionally, it calculates $A_{V_i} = h(ID_{V_i} \parallel PW^l_{V_i} \parallel R_{V_i})$, and $K_{V_i} = A^1_{V_i} \oplus A^2_{V_i}$, $AD^l = rn_{V_i}$, $Nn^l = A^2_{V_i}$. In addition, by using ASCON decryption function, the plaintext PT_{V_i}, which is equal to $PT_{V_i} = \{PID_{V_i}, s_{V_i}\}$, and authentication parameter $MAC^l_{V_i}$ are retrieved as $(PT_{V_i}, MAC^l_{V_i}) = \mathscr{D}_{K_{V_i}}\{(AD^l, Nn^l), CT_{V_i}\}$. Further, V_i checks $MAC^l_{V_i} \overset{?}{=} MAC_{V_i}$. If so, the vehicle user is successfully logged into V_i. Now VO_i can change/reset the password.

PR-3: VO_i inputs the new password $PW^{new}_{V_i}$ into the smart application. Further, the smart application and the corresponding OBU_{V_i} calculate $A^{new}_{V_i} = h(ID_{V_i} \parallel PW^{new}_{V_i} \parallel R_{V_i})$, $K^{new}_{V_i} = A^{new^1}_{V_i} \oplus A^{new^2}_{V_i}$, $AD = rn_{V_i}$, and $Nn = A^{new^2}_{V_i}$. Moreover, using ASCON encryption, OBU_{V_i} computes $(CT^{new}_{V_i}, MAC^{new}_{V_i}) = \mathscr{E}_{K^{new}_{V_i}}\{(AD, Nn), PT_{V_i}\}$, where $PT_{V_i} = \{PID_{V_i}, s_{V_i}\}$.

PR-4: Finally, smart application keeps the updated parameters $\{CT^{new}_{V_i}, MAC^{new}_{V_i}, rn_{V_i}, C_{V_i}, PUF(\cdot), Q_{V_i}\}$ in the OBU_{V_i} of V_i.

G. Block creation, verification and addition phase

This subsection provides a comprehensive overview of the block construction, verification, and addition procedure employed within the designed framework. In this phase, the process of securely sending data as a transaction to the CSN via an RSU is initiated. Transactions are generated when vehicles exchange information related to various events, such as traffic conditions, hazardous road conditions, or any other relevant data.

Upon generation, these transactions are propagated to the CSN through the RSUs, where they are collected in the transactions pool. The transactions pool serves as a temporary repository, allowing peer CSs within the network to access and validate the incoming transactions. Each CS actively maintains and updates its copy of the transactions pool to ensure that the most recent data is available for processing.

To achieve consensus and guarantee the security and reliability of the blockchain, the framework elects a leader in a round-robin fashion from among the peer CSs when the transactions pool reaches a certain limit. The leader plays a crucial role in the block creation process. Once chosen, the leader constructs a block by aggregating a set of validated transactions from the transactions pool. The block includes the transactions, a reference to the previous block, a timestamp, and other necessary metadata, as illustrated in Fig. 4.

To validate the proposed block and achieve consensus, the leader employs the "Practical Byzantine Fault Tolerance (PBFT)" consensus process [43]. PBFT is a well-established voting-based consensus algorithm known for its ability to tolerate up to a certain number of malicious actors (Byzantine faults) within the network. The PBFT process involves a series of message exchanges among the leader and follower CSs to reach a consensus on the acceptance of the proposed block.

Block Header	
Block version	BV
Last hash	$PHash$
Merkle root hash	$MRHash$
Timestamp	TS
Proposer ID	ID of CS_x
Proposer's public key	PB_{CS_x}
Data (Encrypted Transactions)	
List of encrypted transactions	$\{(Tx_i)\mid i = 1, 2, \cdots, n_t)\}$
Current block hash	$CHash$
Signature on $CHash$	$ECDSA.Sign(CHash)$

Fig. 4: Block $Block_m$ Composition.

After the PBFT process achieves consensus successfully, the newly formed block becomes valid and is eligible for incorporation into the blockchain. The designated leader triggers the transmission of a commit message to the follower nodes, indicating that the block has been approved and can be included in their individual blockchains. Upon receiving the commit message, the follower nodes verify the block's integrity and add it to their local copies of the blockchain. To maintain network-wide consistency, the followers then broadcast the commit messages to the entire network, informing all nodes of the block's inclusion.

The block addition process, underpinned by the PBFT consensus mechanism, guarantees the immutability and tamper resistance of the blockchain. This robust approach ensures that the data exchanged among vehicles is reliably stored and becomes an integral part of the transparent and permanent ledger maintained by the CSN. Moreover, the voting-based consensus enhances the security of the system, making it resilient against potential malicious attacks and ensuring the overall stability and trustworthiness of the blockchain-based vehicular network environment.

V. SECURITY ANALYSIS OF THE PROPOSED BASF-ITS

This section presents a comprehensive security analysis revealing the robustness of BASF-ITS against numerous potential cyber security assaults.

1) Replay attack

In BASF-ITS, we address the issue of replay attacks to ensure the security of communications between different entities. During the AKA phase between neighboring vehicles, denoted as V_i and V_j (as discussed in Section IV-D1), the messages $msg_{VV_1} : \{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ and $msg_{VV_2} : \{CT_2, MAC_3, TS_{V_j}\}$ are exchanged over insecure channels. Likewise, during the AKA phase between the CH and its associated RSU, denoted as V_i and RSU_k (as described in Section IV-D2), the messages $msg_{VR_1} : \{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ and $msg_{VR_2} : \{CT_2, MAC_3, TS_{RSU_k}\}$ are communicated over public channels. To prevent replay attacks in BASF-ITS, we employ random numbers and current timestamps to construct the AKA messages. This ensures that the messages exchanged are fresh and have not been captured and re-transmitted by an adversary. When an adversary, denoted as $\mathscr{A}$, attempts to replay old messages, they are easily detected at the receiving end. This is because the BASF-ITS implementation verifies the freshness of the received messages, making it inherently resistant to replay attacks.

2) *MitM attack*

In BASF-ITS, we have implemented measures to ensure the resilience of our framework against MitM attacks. MitM attacks involve an adversary, referred to as $\mathscr{A}$, attempting to disrupt the communication channel between vehicles, particularly between V_i and V_j. To execute the MitM attack, $\mathscr{A}$ aims to eavesdrop on the AKA request message msg_{VV_1} exchanged over the insecure channel and then alter it to masquerade as a legitimate vehicle within the network. However, our framework's design makes it computationally challenging for $\mathscr{A}$ to achieve success due to the following factors: i) selecting the correct timestamp and random nonce, and ii) the need for secret key s_{V_i}. Similarly, $\mathscr{A}$ cannot construct the acknowledgment AKA message msg_{VV_2}. Therefore, these inherent complexities and dependencies on secret keys and parameters render the MitM attack computationally difficult for $\mathscr{A}$ to execute successfully.

3) *Anonymity and untraceability preservation*

The anonymity of the vehicle is a crucial security trait of an AKA scheme as it serves two goals. The first goal is to ensure that the vehicle's privacy is protected, which means that the adversary cannot determine the vehicle's real identity. The second goal is to ensure that the vehicle is untraceable, which means that $\mathscr{A}$ cannot correlate two AKA sessions carried out by the same vehicle. In BASF-ITS, $\mathscr{A}$ cannot determine the vehicle's real identity based on the transmitted messages during the V2V AKA phase. To acquire the vehicle's real identity, $\mathscr{A}$ must know the secret credentials, i.e., private keys of the vehicles and random nonces. However, it is computationally hard in polynomial time for $\mathscr{A}$ to obtain the identities of the vehicles. Therefore, BASF-ITS guarantees the anonymity trait. Moreover, the exchanged messages during the V2V AKA phase, i.e., $msg_{VV_1} : \{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ and $msg_{VV_2}\{CT_2, MAC_3, TS_{V_j}\}$ are generated using current timestamps and random numbers for every new AKA session. Thus, msg_{VV_1} and msg_{VV_2} are unique for each session, making it challenging for $\mathscr{A}$ to link the intercepted messages from different AKA sessions. Consequently, $\mathscr{A}$ faces significant difficulty in tracing vehicles during the V2V AKA phase of BASF-ITS. Therefore, BASF-ITS also preserves the untraceability trait. Similarly, the AKA scheme of BASF-ITS between a vehicle and its associated RSU also ensures anonymity and untraceability traits.

4) *Resilience against vehicle capture attack*

If adversary $\mathscr{A}$ gets a lost or stolen vehicle, it can extract the parameters $\{CT_{V_i}, MAC_{V_i}, rn_{V_i}, C_{V_i}, PUF(\cdot), Q_{V_i}\}$ stored in OBU of the vehicle by executing power analysis attacks. From the extracted parameters, yet, $\mathscr{A}$ cannot acquire secret credentials, like PID_{V_i} and s_{V_i}. To obtain these secret credentials, $\mathscr{A}$ requires to compute $R_{V_i} = PUF(C_{V_i})$, $A_{V_i} = h(ID_{V_i} \parallel PW^l_{V_i} \parallel R_{V_i})$, $K_{V_i} = A^1_{V_i} \oplus A^2_{V_i}$, $AD^l = rn_{V_i}$ $Nn^l = A^2_{V_i}$, and $(PT_{V_i}, MAC^l_{V_i}) = \mathscr{D}_{K_{V_i}}\{(AD^l, Nn^l), CT_{V_i}\}$. Nonetheless, for these computations to be performed, $\mathscr{A}$ necessitates the secret parameter PW_{V_i}, which remains exclusive to the vehicle owner. As a result, our BASF-ITS exhibits robustness against vehicle capture attacks.

5) *ESL attack*

In BASF-ITS, secure communication between vehicles and associated RSUs is established through the use of shared session keys. In this context, we consider the ESL attack, and the measures taken to ensure the security of the session keys. During the AKA phase between vehicles V_i and V_j, a secret shared session key $SK_{V_jV_i}$ is derived. This key is computed as follows: $SK_{V_iV_j}(=SK_{V_jV_i}) = h((s_{V_i} \cdot s_{V_j} \cdot P) \parallel PID_{V_i} \parallel PID_{V_j} \parallel (r_{V_i} \cdot Q_{V_j}) \parallel r_{V_j} \parallel TS_{V_i} \parallel TS_{V_j})$. Similarly, during the AKA phase between the CH V_i and the associated RSU RSU_k, another secret shared session key SK_{V_i,RSU_k} is created for secure communication. The computation for this key is given as: $SK_{V_i,RSU_k}(= SK_{RSU_k,V_i}) = h((s_{V_i} \cdot s_{RSU_k} \cdot P) \parallel PID_{V_i} \parallel PID_{RSU_k} \parallel (r_{V_i} \cdot P) \parallel r_{RSU_k} \parallel TS_{V_i} \parallel TS_{RSU_k})$. To protect against the ESL attack, the CK-adversary model is employed, as described in Subsection III-B. In this model, an adversary $\mathscr{A}$ may have access to short-term secrets, such as r_{V_i}, r_{V_j}, TS_{V_i}, and TS_{V_j}. However, to produce the session key, the adversary also requires knowledge of the long-term secrets, specifically s_{V_i} and s_{V_j} for the participating vehicles. Obtaining these long-term secrets poses a computationally hard challenge for the adversary, ensuring the security of the session key. The same level of security is maintained for the session key between CH V_i and associated RSU RSU_k since the adversary would need the long-term secrets s_{V_i} and s_{RSU_k} to compromise the session key. However, the distinctness and randomness of each session key prevent the exposure of past and future session keys even if the current one is compromised. The protection of both backward and forward secrecy, along with the security of the session keys, allows BASF-ITS to effectively defend against the ESL attack. This resilience ensures the integrity and confidentiality of communications within the ITS environment.

6) *DoS attack*

The vehicle driver enters his/her password into the smart application running on the vehicle (for instance, V_i) at the time of the user login phase. The smart application then computes $R_{V_i} = PUF(C_{V_i})$, $A_{V_i} = h(ID_{V_i} \parallel PW^l_{V_i} \parallel R_{V_i})$, $K_{V_i} = A^1_{V_i} \oplus A^2_{V_i}$, $AD^l = rn_{V_i}$, $Nn^l = A^2_{V_i}$ and $(PT_{V_i}, MAC^l_{V_i}) = \mathscr{D}_{K_{V_i}}\{(AD^l, Nn^l), CT_{V_i}\}$. Next, OBU_{V_i} verifies the condition $MAC^l_{V_i} \overset{?}{=} MAC_{V_i}$. If the condition becomes valid, the vehicle driver can accomplish future tasks. The login operation in BASF-ITS is solely processed on the V_i side without involving other entities like vehicles or RSUs. This characteristic ensures its resilience against DoS attacks. Additionally, as the login process doesn't consume additional bandwidth from other ITS components, potential attackers are unable to overwhelm the system, guaranteeing uninterrupted access for authorized users. BASF-ITS remains robust against DoS attacks, ensuring smooth functioning and availability for legitimate users.

7) *Impersonation attacks*

Suppose an adversary $\mathcal{A}$ behaves as a legitimate vehicle (for instance, V_i) to the neighbor vehicle V_j. In such circumstances, $\mathcal{A}$ may then try to construct a legitimate AKA request message msg_{VV_1} : $\{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ to impersonate V_i. $\mathcal{A}$ can begin by selecting a timestamp $TS^*_{V_i}$ and a random secret $r^*_{V_i}$ to achieve this goal. Subsequently, $\mathcal{A}$ may attempt to compute $RB_{V_i} = r^*_{V_i} \cdot P$, $RSC_{V_iV_j} = r^*_{V_i} \cdot Q_{V_j}$, $B_{V_i} = h(RSC_{V_iV_j} \parallel TS^*_{V_i})$, $K_1 = B^1_{V_i} \oplus B^2_{V_i}$, and $(CT_1, MAC_1) = \mathcal{E}_{K_1}\{(B^2_{V_i}, B^2_{V_i}), (PID_{V_i} \parallel Q_{V_i})\}$ and then construct msg_{VV_1}. Nonetheless, in order to generate a valid and authentic message, $\mathcal{A}$ needs access to the random secret r_{V_i}, which is exclusive to the genuine vehicle V_i. As a result, $\mathcal{A}$ cannot successfully impersonate V_i. Similarly, $\mathcal{A}$ is unable to impersonate an RSU since it lacks the necessary long-term secrets. Therefore, BASF-ITS effectively withstands impersonation attacks, ensuring the security and authenticity of V2V and V2RSU communications.

VI. BLOCKCHAIN IMPLEMENTATION

This section briefly explains how blockchain is implemented within the proposed BASF-ITS framework by creating a virtual distributed system with the aid of Node.js scripts. The scripts were developed using Visual Studio Code (version 1.60) integrated development environment and with the following system configuration: “Ubuntu 16.04 LTS, with 8 GiB memory, Intel® Core ™ i7-6700, CPU @ 3.4 GHz, and 64-bit OS type.”

In this study, we make several key assumptions about the CSN and its block structure. First, we assume the existence of 13 cloud servers within the CSN, forming a fully connected structure. The block structure is depicted in Fig. 4, and it serves as the basis for our analysis. Moreover, to determine the block size, we consider various components and their respective bit sizes: block version (32 bits), previous block hash (256 bits), merkle tree root (256 bits), proposer identity (160 bits), public key of the proposer (320 bits), timestamp (32 bits), block payload ($640 \cdot n_t$ bits, where n_t denotes the number of stored transactions per block), current block hash (SHA-256, 256 bits), and “elliptic curve digital signature algorithm (ECDSA)” signature (320 bits). Furthermore, each transaction is encrypted using ECC encryption, resulting in two elliptic curve points that collectively demand 640 bits. Considering all these factors, the total size of a block becomes $1632 + 640 \cdot n_t$ bits. For the simulation, we utilize a voting-based PBFT consensus algorithm. The simulation encompasses two distinct cases, each of which will be thoroughly explored and analyzed in our research.

Case 1: This scenario involves the pragmatic study of the proposed BASF-ITS, where we analyze a fixed number of transactions (35 per block) while varying the number of mined blocks in the blockchain at different times. Fig. 5 illustrates the relationship between the overall computing time (in seconds) and the number of mined blocks, showcasing a linear correlation.

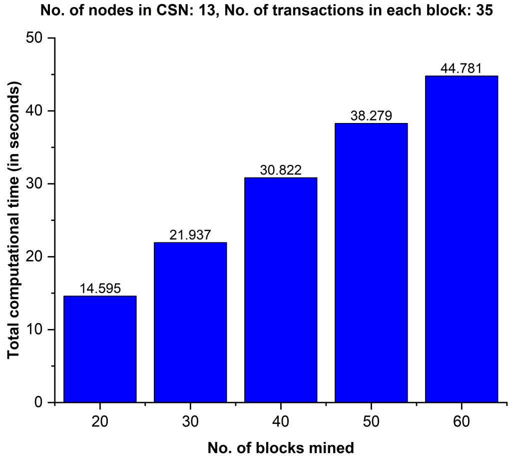

Fig. 5: Findings from the blockchain simulation: Case 1.

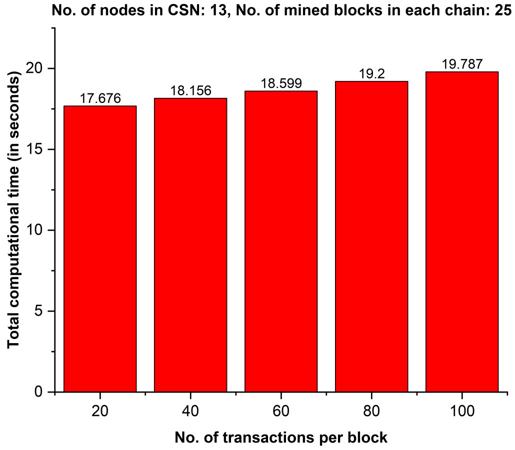

Fig. 6: Findings from the blockchain simulation: Case 2.

Case 2: For this scenario, we establish a constant number of mined blocks at 25 for each chain. The simulation results, presented in Fig. 6, reveal that the overall computational time (in seconds) exhibits a linear correlation with the number of transactions stored in a block, with the chain length fixed.

Remark 4. *In our pragmatic study, we have explored two types of computational times: Case 1: We measured the computational time required for mining a varied number of blocks (20, 30, 40, 50, and 60) into the blockchain. Each block in this case contains a fixed number of transactions (35). Case 2: We assessed the computational time needed for mining a fixed number of blocks (25) into the blockchain. However, in this scenario, each block contains a varied number of transactions (20, 40, 60, 80, and 100). To estimate the computational time in both cases, we considered the summation of the following factors according to the PBFT consensus algorithm: (i) the time required to establish a socket connection between two P2P server nodes. (ii) the time spent on generating and broadcasting different types of messages (e.g., “TRANSACTION,” “PREPARE,” “PRE-PREPARE,” “COMMIT,” and “ROUND*

TABLE II: Execution time of cryptography operations

Operation	Scenario 1: Raspberry PI-3 (ms)	Scenario 2: Server (ms)
T_{as}	0.370	0.0351
T_{bpo}	8.123	4.42
T_{eca}	0.124	0.006
T_{ecm}	2.850	0.780
$T_{fe} \approx T_{ecm}$	2.850	0.780
T_{ma}	0.010	0.001
T_{exp}	1.42	0.042
T_{mul}	0.011	0.002
T_{mtp}	0.385	0.114
T_{puf}	0.59 μs	-
T_h	0.345	0.039
T_{se}/T_{sd}	0.391	0.02

CHANGE”). (iii) the time needed to build the transaction pool, wallet (private and public key generation), prepare pool, block pool, commit pool, and messages pool. and (iv) the time required to add the message and the block to the appropriate message pool and block pool, as well as to append new blocks into the blockchain. Although the experimental blockchain simulations in this article were carried out using “Python” and “Node.js” programming languages, yet a system having high computational capabilities may reduce computation time. Moreover, the simulation did not consider the P2P network’s link delay, byzantine ratio, or other network traits. Our future work will seek to consider these parameters.

VII. COMPARATIVE ANALYSIS

This section provides a comprehensive comparative analysis of BASF-ITS with the schemes proposed by Chattaraj *et al.* [28], Vangala *et al.* [29], Ever [30], Ali *et al.* [31], and Wazid *et al.* [36]. Our evaluation encompasses various aspects, including security and functionality features, as well as computational, communication, and storage costs during the AKA phase. These evaluation metrics are fundamental in determining the efficiency and scalability of our proposed BASF-ITS scheme, offering valuable insights into the computational complexity, data exchange, and storage space requirements involved in the AKA processes. Notably, we exclude the overheads related to registration and password reset procedures from this analysis, given their infrequent occurrence. To facilitate a deeper understanding of the comparison, we present our findings in detail within the subsequent subsections.

A. Computation cost

This paper considers the experimental results reported in [44] and [45] for numerous cryptographic primitives and operations to calculate the computational cost of BASF-ITS and other competing schemes. The execution times on different platforms for numerous cryptographic primitives are given in Table II. We denote T_{as}, T_{bpo}, T_{eca}, T_{ecm}, $T_{fe} \approx T_{ecm}$, T_{ma}, T_{exp}, T_{mul}, T_{mtp}, T_{puf}, T_h, and T_{se}/T_{sd} as the time required for ASCON, bilinear pairing, ECC point addition, ECC point multiplication, fuzzy extractor function, modular addition, modular exponentiation, modular multiplication, map-to-point, $PUF(\cdot)$, SHA-256 hash function, and symmetric encryption or decryption, respectively. Moreover, we discard the time required to calculate bitwise XOR operation as it is negligible. Further, in Table II, Scenario-1 is considered for resource-limited devices, i.e., IoT sensors, sensing devices, etc., utilizing the setting: “Raspberry PI-3 (R-PI3), Ubuntu 16.04 LTS, OS 64- bits, 1.2 GHz Quad-core processor, and RAM 1 GiB”. Conversely, Scenario-2 is considered for resource-high devices, i.e., RSUs, gateway nodes, servers, etc., utilizing the setting: “Ubuntu 16.04 LTS, with 8 GiB memory, Intel® Core ™ i7-6700, CPU @ 3.4 GHz, and 64-bit OS type.”.

For our proposed BASF-ITS, in V2CH case, a smart vehicle V_i and its associated CH V_j demand the computational costs of $2T_{as} + 3T_{ecm} + 4T_h$ and $2T_{as} + 2T_{ecm} + 4T_h$, respectively. Therefore, the total computational cost for V2CH case is $4T_{as} + 5T_{ecm} + 8T_h \approx 18.49$ ms. Similarly, for the CH2RSU case, the CH V_j and its associated RSU RSU_j demand the computational costs of $2T_{as} + 3T_{ecm} + 4T_h \approx 10.67$ ms and $3T_{as} + 2T_{ecm} + 5T_h + T_{puf} \approx 1.8603$ ms, respectively. Hence, the total computational cost of CH2RSU case demands $5T_{as} + 5T_{ecm} + 9T_h + T_{puf} \approx 12.5303$ ms. The computation costs of BASF-ITS (V2CH and CH2RSU) and other competing schemes are compared and summarized in Table III, demonstrating that BASF-ITS requires less computation costs

TABLE III: Comparison of computation cost of BASF-ITS and comparable schemes

Scheme	OBU/Vehicle/CH	RSU/Server/CS
Chattaraj *et al.* [28] (V2CH)	$2(T_{eca} + 4T_h + 5T_{ecm}) \approx 31.508$ ms	—
Chattaraj *et al.* [28] (CH2RSU)	$5T_h + 5T_{ecm} + T_{eca} \approx 16.099$ ms	$3T_h + 5T_{ecm} + T_{eca} \approx 4.023$ ms
Chattaraj *et al.* [28] (RSU2CS)	—	$2(T_{poly} + 6T_h) \approx 6.468$ ms
Vangala *et al.* [29] (V2CH)	$2(5T_h + 2T_{eca} + 6T_{ecm}) \approx 38.146$ ms	—
Vangala *et al.* [29] (CH2RSU)	$7T_h + 2T_{eca} + 6T_{ecm} \approx 19.763$ ms	$8T_h + 2T_{eca} + 6T_{ecm} \approx 5.004$ ms
Ever [30]	$2T_{bpo} + 9T_h + 3T_{ecm} + 2T_{mtp} \approx 28.671$ ms	$3T_{bpo} + 6T_h + 3T_{ecm} + 2T_{mtp} \approx 16.062$ ms
Ali *et al.* [31]	$T_{se} + 18T_h + T_{fe} \approx 9.451$ ms	$3T_{se}/T_{sd} + 7T_h \approx 0.333$ ms
Wazid *et al.* [36] (V2CH)	$11T_{ecm} + 18T_h \approx 37.56$ ms	—
Wazid *et al.* [36] (CH2RSU)	$5T_{ecm} + 8T_h \approx 17.01$ ms	$5T_{ecm} + 7T_h \approx 4.173$ ms
BASF-ITS (V2CH)	$5T_{ecm} + 8T_h + 4T_{as} \approx 18.49$ ms	—
BASF-ITS (CH2RSU)	$3T_{ecm} + 4T_h + 2T_{as} \approx 10.67$ ms	$2T_{ecm} + 5T_h + 3T_{as} + T_{puf} \approx 1.8603$ ms

Note: In the scheme proposed by Chattaraj $et\ al.$ [28], a t-degree polynomial necessitates t modular multiplications and t modular additions, denoted as $T_{poly} = tT_{mul} + tT_{ma}$. Here, we assume $t = 1000$.

TABLE IV: Comparison of communication costs

Scheme	**Number of messages**	**Total cost (bits)**
Chattaraj *et al.* [28] (V2CH)	3	2464
Chattaraj *et al.* [28] (CH2RSU)	3	2560
Chattaraj *et al.* [28] (RSU2CS)	3	1376
Vangala *et al.* [29] (V2CH)	2	1856
Vangala *et al.* [29] (CH2RSU)	3	2400
Ever [30]	6	5344
Ali *et al.* [31]	3	3424
Wazid *et al.* [36] (V2CH)	3	2208
Wazid *et al.* [36] (CH2RSU)	3	2016
BASF-ITS (V2CH)	2	1504
BASF-ITS (CH2RSU)	2	1504

TABLE V: Comparison of security and functionality features

Features	**Chattaraj** *et al.* [28]	**Vangala** *et al.* [29]	**Ever** [30]	**Ali** *et al.* [31]	**Wazid** *et al.* [36]	**BASF-ITS**
$\mathcal{FE}_1$	✓	✓	✓	✓	×	✓
$\mathcal{FE}_2$	✓	✓	✓	✓	×	✓
$\mathcal{FE}_3$	✓	✓	✓	✓	✓	✓
$\mathcal{FE}_4$	✓	✓	✓	✓	✓	✓
$\mathcal{FE}_5$	✓	✓	✓	✓	✓	✓
$\mathcal{FE}_6$	✓	✓	✓	✓	✓	✓
$\mathcal{FE}_7$	✓	✓	×	×	✓	✓
$\mathcal{FE}_8$	✓	✓	✓	✓	✓	✓
$\mathcal{FE}_9$	×	×	×	×	✓	✓
$\mathcal{FE}_{10}$	✓	✓	×	×	✓	✓
$\mathcal{FE}_{11}$	✓	✓	×	✓	✓	✓

Note: ✓: denotes the availability of features; × : indicates the feature not available

than the other relevant state-of-the-art schemes with the exception of the scheme of Ali *et al.* [31]. Despite requiring higher computational cost compared to the scheme proposed by Ali $et\ al.$ (2020) [31], our scheme provides a broader range of functionality and enhanced security features (refer to Table V).

B. Communication cost

To estimate the communication cost of our proposed BASF-ITS, we have made the following assumptions in order to calculate the communication cost. We consider the sizes in bit-length of random numbers, pseudo-identities, associative data, authentication parameters, timestamps, hash function, and elliptic curve points to be $128, 128, 128, 128, 32, 256,$ and 320 bits, respectively. In the V2CH case of BASF-ITS, the communication costs for two messages msg_{VV_1} : $\{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ and msg_{VV_2} : $\{CT_2, MAC_3, TS_{V_j}\}$ demand $(448 + 128 + 320 + 32) = 928$ bits and $(416 + 128 + 32) = 576$ bits, respectively, which altogether need 1504 bits. Similarly, in the CH2RSU case of BASF-ITS, the communication costs for two messages msg_{VR_1} : $\{CT_1, MAC_1, RB_{V_i}, TS_{V_i}\}$ and msg_{VR_2} : $\{CT_2, MAC_3, TS_{RSU_k}\}$ demand $(448 + 128 + 320 + 32) = 928$ bits and $(416 + 128 + 32) = 576$ bits, respectively, which altogether need 1504 bits. The communication costs of BASF-ITS (V2CH and CH2RSU) and other competing schemes are compared and summarized in Table IV, demonstrating that BASF-ITS requires less communication costs than the other state-of-the-art schemes.

C. Storage cost

The proposed BASF-ITS framework stores a total of five authentication-related parameters, namely $\{CT_{V_i}, MAC_{V_i}, rn_{V_i}, C_{V_i}, Q_{V_i}\}$, in addition to the functions $h(\cdot)$ and $PUF(\cdot)$, as well as the system parameters $\{GF, P\}$. Notably, in all competing authentication schemes, the system parameters and functions necessitate minimal memory and are stored within the smart card. Hence, our analysis and comparison efforts are centered solely on the storage aspects of authentication-related parameters. The storage cost of our proposed BASF-ITS framework, which includes the parameters $\{CT_{V_i}, MAC_{V_i}, rn_{V_i}, C_{V_i}, Q_{V_i}\}$ is calculated as follows: $\{288 + 128 + 128 + 128 + 320\} = 992$ bits. In comparison, the storage costs (in bits) of other authentication schemes are as follows: Chattaraj *et al.* [28]: 960 bits, Vangala *et al.* [29]: 640 bits, Ever [30]: 800 bits, Ali *et al.* [31]: 800 bits, and Wazid *et al.* [36]: 1280 bits. To provide a clear comparison, the storage costs of our proposed BASF-ITS framework and the competing schemes are presented in Table VI. The results clearly demonstrate that BASF-ITS outperforms Wazid *et al.* [36] in terms of storage efficiency. It is worth mentioning that the higher storage cost of BASF-ITS is justified by the fact that it offers enhanced security and functionality features, as indicated in Table V.

TABLE VI: Comparison of storage costs

Scheme	**Cost (bits)**
Chattaraj *et al.* [28]	960
Vangala *et al.* [29]	640
Ever [30]	800
Ali *et al.* [31]	800
Wazid *et al.* [36]	1280
BASF-ITS	992

D. Security and functionality features comparison

Table V summarises the comparison of the proposed BASF-ITS and the other competing schemes based on the set of eleven functionality and security features, namely, $\mathcal{FE}_1$: mutual authentication; $\mathcal{FE}_2$: key agreement; $\mathcal{FE}_3$: resilience against device (vehicle/RSU) physical capture attack; $\mathcal{FE}_4$: replay attack; $\mathcal{FE}_5$: MitM attack; $\mathcal{FE}_6$: impersonation attacks; $\mathcal{FE}_7$: ESL attack; $\mathcal{FE}_8$: DoS attack; $\mathcal{FE}_9$: anonymity and untraceability preservation; $\mathcal{FE}_{10}$: support blockchain solution; and $\mathcal{FE}_{11}$: dynamic node addition phase. It is worth mentioning that our proposed BASF-ITS and the schemes of Chattaraj *et al.* [28], Vangala *et al.* [29], and Wazid *et al.* [36] support blockchain solutions. However, BASF-ITS renders more functionality and higher security.

The results of this section reveal the superior performance of BASF-ITS, owing to its adept incorporation of various efficient cryptographic primitives. BASF-ITS strategically amalgamates hash functions, XOR operator, ASCON, ECC, and PUF, which collectively play a pivotal role in fortifying the security and functionality features of the proposed AKA schemes. This combination not only enhances security but also bolsters the efficiency of the system. Moreover, the integration of blockchain technology within BASF-ITS acts as a robust safeguard, ensuring the protection of data at rest from potential tampering attempts. These design strategies, in conjunction with blockchain utilization, contribute significantly to heightened security and efficiency. Consequently, BASF-ITS surpasses competing schemes in communication, computation, and storage costs, showcasing its superior performance and resilience.

E. Critical discussion

Within the ITS environment, multiple entities engage in communication through public channels, rendering them susceptible to a wide range of security vulnerabilities. In response to these critical security concerns, our proposed framework, BASF-ITS, emerges as a robust solution specifically designed to safeguard data during transit within the ITS environment. This is achieved by amalgamating efficient cryptographic primitives, such as hash functions, XOR operator, ASCON, ECC, and PUF, into our AKA schemes, showcasing its resilience against potential security threats. The incorporation of the PUF trait provides an additional layer of security, safeguarding against physical attacks on smart vehicles and RSUs. In addition to the transit security, our framework strategically ensures data integrity and protection on cloud servers through the application of blockchain technology. The immutable and decentralized nature of the blockchain effectively shields data at rest from any tampering attempts. The BASF-ITS framework presents a comprehensive and reliable approach to fortify the overall system security during both data transit and storage, making it an ideal choice for data protection in smart vehicular environments. Our research demonstrates the practicality of deploying the proposed solution as a robust tool to efficiently address the security challenges within the ITS domain. Additionally, the proposed framework stands out due to its lightweight and efficient characteristics, enabling facile deployment in various ITS applications and other resource-constrained environments.

BASF-ITS can be easily implemented as a robust tool for securing ITSs in the real world. Nevertheless, it is imperative to acknowledge that for the proposed framework to be operational, the participating entities, including vehicles and roadside units, must possess PUF-enabled capabilities. Our future work includes conducting a dedicated investigation into the sensitivities of key parameters in our proposed scheme to gain deeper insights into its behavior and performance under varied conditions, contributing to a more comprehensive understanding of its adaptability, robustness, and limitations. Furthermore, we recognize that systems equipped with higher computational capabilities hold the potential to reduce computation time, making this aspect critical for consideration in real-world deployment scenarios. As part of our future work, we intend to encompass specific network parameters, such as P2P network link delay, byzantine ratio, and other network traits, in our simulations. These parameters significantly influence the overall performance of a blockchain system. By addressing these factors, our future work aims to provide a more comprehensive evaluation of BASF-ITS under diverse network conditions.

VIII. CONCLUSION AND FUTURE WORK

This article has proposed a blockchain-assisted lightweight authenticated key agreement security framework for smart vehicles-enabled intelligent transportation system, called BASF-ITS. BASF-ITS effectively addresses the critical issue of data security during transit through its incorporation of cleverly combined cryptographic primitives, including hash functions, XOR operator, ASCON, elliptic curve cryptography, and physical unclonable function (PUF). The integration of the PUF trait emerges as a pivotal strength, empowering smart vehicles and RSUs to proactively defend against physical attacks and thwart tampering attempts, significantly enhancing the overall system security. Our successful integration of blockchain technology into the framework provides a robust safeguard against tampering attempts on data at rest when stored on cloud servers, adding an essential layer of security and further fortifying the reliability and strength of BASF-ITS. Theoretical analysis verifies that BASF-ITS demonstrates resilience against various potential security attacks. Moreover, BASF-ITS offers enhanced security and additional functionality traits, such as anonymity, untraceability, and physical security. Notably, the framework exhibits exceptional performance in resource-constrained environments, surpassing numerous relevant state-of-the-art schemes in terms of computation, communication, and storage costs. In future work, we aim to devise a security and privacy-aware handover authentication scheme for a blockchain-enabled intelligent transportation system to address the reauthentication problem in the IoV environment. Additionally, we will configure pragmatic network scenarios to evaluate the performance in terms of throughput and average delay, further validating the effectiveness and applicability of BASF-ITS in real-world settings.

REFERENCES

[1] H. Tao, J. Qiu, Y. Chen, V. Stojanovic, and L. Cheng, "Unsupervised cross-domain rolling bearing fault diagnosis based on time-frequency information fusion," *J. Frank. Inst.*, vol. 360, no. 2, pp. 1454–1477, Jan. 2023.

[2] Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, and W. Paszke, "An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints," *IEEE Trans. Syst. Man Cybern.: Syst.,* vol. 53, no. 6, pp. 3461–3473, June 2023.

[3] H. Tao, L. Cheng, J. Qiu, and V. Stojanovic, "Few shot cross equipment fault diagnosis method based on parameter optimization and feature mertic," *Meas. Sci. Technol.,* vol. 33, no. 11, p. 115005, Aug. 2022.

[4] X. Song, P. Sun, S. Song, and V. Stojanovic, "Quantized neural adaptive finite-time preassigned performance control for interconnected nonlinear systems," *Neural. Comput. Appl.,* vol. 35, no. 21, pp. 15429–15446, Apr. 2023.

[5] X. Song, C. Wu, V. Stojanovic, and S. Song, "1 bit encoding-decoding-based event-triggered fixed-time adaptive control for unmanned surface vehicle with guaranteed tracking performance," *Control Eng. Pract.,* vol. 135, p. 105513, Jun. 2023.

[6] X. Song, N. Wu, S. Song, and V. Stojanovic, "Switching-like event-triggered state estimation for reaction-diffusion neural networks against DOS attacks," *Neural Process. Lett.,* vol. 55, no. 7, pp. 8997–9018, Mar. 2023.

[7] S. Yu, *et al.*, "IoV-SMAP: Secure and efficient message authentication protocol for IoV in smart city environment," *IEEE Access*, vol. 8, pp. 167875–167886, Sep. 2020.

[8] T. Limbasiya and D. Das, "IoVCom: Reliable comprehensive communication system for Internet of Vehicles," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 6, pp. 2752–2766, Dec. 2019.

[9] M. Shen, *et al.*, "Secure and efficient blockchain-assisted authentication for edge-integrated Internet-of-Vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12250–12263, Nov. 2022.

[10] A. Badshah, *et al.*, "A novel framework for smart systems using blockchain-enabled Internet of Things," *IT Prof.*, vol. 24, no. 3, pp. 73–80, June 2022.

[11] A. Hammoud, *et al.*, "AI, blockchain, and vehicular edge computing for smart and secure IoV: Challenges and directions," *IEEE Internet of Things Mag.*, vol. 3, no. 2, pp. 68–73, June. 2020.

[12] S. Abbas, *et al.*, "Blockchain-based authentication in Internet of Vehicles: A survey," *Sensors*, vol. 21, no. 23, pp. 7927, Nov. 2021.

[13] M. A. Ferrag, *et al.*, "Blockchain technologies for the Internet of Things: Research issues and challenges," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 2188–2204, Mar. 2018.

[14] S. Kumar, *et al.*, "A survey on the blockchain techniques for the Internet of Vehicles security," in press, *Trans. Emerg. Telecommun. Technol.*, pp. e4317, June 2021.

[15] P. Bagga, *et al.*, "Authentication protocols in Internet of Vehicles: Taxonomy, analysis, and challenges," *IEEE Access*, vol. 8, pp. 54314–54344, Mar. 2020.

[16] M. B. Mollah, *et al.*, "Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey," *IEEE Internet of Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021.

[17] D. S. Gupta, A. Karati, W. Saad, and D. B. da Costa, "Quantum-defended blockchain-assisted data authentication protocol for Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3255–3266, Jan. 2022.

[18] S. Roy *et al.*, "Blockchain-based efficient access control with handover policy in IoV-enabled intelligent transportation system," *IEEE Trans. Veh. Technol.*, early access, pp. 1–16, Oct. 2023.

[19] Q. Xie, Z. Ding, W. Tang, D. He, and X. Tan, "Provable secure and lightweight blockchain-based V2I handover authentication and V2V broadcast protocol for VANETs," *IEEE Trans. Veh. Technol.*, early access, pp. 1–12, Jun. 2023.

[20] Y. Liu, D. He, M. Luo, H. Wang, and Q. Liu, "ATRC: An anonymous traceable and revocable credential system using blockchain for VANETs," *IEEE Trans. Veh. Technol.*, early access, pp. 1–14, Sep. 2023.

[21] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2467–2476, Sep. 2017.

[22] M. Azees, *et al.*, "BBAAS: Blockchain-based anonymous authentication scheme for providing secure communication in VANETs," *Secur. Commun. Netw.*, vol. 2021, no. 6679882, Feb. 2021.

[23] P. K. Sadhu, J. Eickholt, V. P. Yanambaka, and A. Abdelgawad, "Supervised machine learning tools and PUF based Internet of Vehicles authentication framework," *Electronics*, vol. 11, no. 23, p. 3845, Nov. 2022.

[24] A. Badshah, *et al.*, "AAKE-BIVT: Anonymous authenticated key exchange scheme for blockchain-enabled Internet of vehicles in smart transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1739–1755, Feb. 2023.

[25] Y. Cho, *et al.*, "A secure three-factor authentication protocol for E-governance system based on multiserver environments," *IEEE Access*, vol. 10, pp. 74351–74365, Jul. 2022.

[26] S. M. Karim, A. Habbal, S. A. Chaudhry, and A. Irshad, "BSDCE-IoV: Blockchain-based secure data collection and exchange scheme for IoV in 5G environment," *IEEE Access*, vol. 11, pp. 36158–36175, Apr. 2023.

[27] N. Xi, W. Li, L. Jing, and J. Ma, "ZAMA: A ZKP-based anonymous mutual authentication scheme for the IoV," *IEEE Internet of Things J.*, vol. 9, no. 22, pp. 22903–22913, Nov. 2022.

[28] D. Chattaraj, *et al.*, "Block-CLAP: Blockchain-assisted certificateless key agreement protocol for Internet of Vehicles in smart transportation," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8092–8107, Aug. 2021.

[29] A. Vangala, *et al.*, "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.

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

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

[32] M. Tanveer, *et al.*, "LAKE-IoD: Lightweight authenticated key exchange protocol for the Internet of Drone environment," *IEEE Access*, vol. 8, pp. 155645–155659, Aug. 2020.

[33] M. Tanveer, G. Abbas, and Z. H. Abbas, "LAS-6LE: A lightweight authentication scheme for 6LoWPAN environments," in *2020 14th International Conference on Open Source Systems and Technologies (ICOSST)* (Lahore, Pakistan), pp. 1–6, Dec. 16-17, 2020.

[34] M. Tanveer, *et al.*, "LAKE-6SH: Lightweight user authenticated key exchange for 6LoWPAN-based smart homes," *IEEE Internet of Things J.*, vol. 9, no. 4, pp. 2578–2591, June. 2021.

[35] M. Tanveer, A. U. Khan, N. Kumar, and M. M. Hassan, "RAMP-IoD: A robust authenticated key management protocol for the Internet of Drones," *IEEE Internet of Things J.*, vol. 9, no. 2, pp. 1339–1353, June. 2021.

[36] M. Wazid, B. Bera, A. K. Das, S. P. Mohanty, and M. Jo, "Fortifying smart transportation security through public blockchain," *IEEE Internet of Things J.*, vol. 9, no. 17, pp. 16532–16545, Sept. 2022.

[37] 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.

[38] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. 2002 International Conference on the Theory and Applications of Cryptographic Techniques, (EUROCRYPT 2002)* (Amsterdam, The Netherlands), pp. 337–351, Apr. 28-May 2, 2002.

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

[40] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. (2014),"ASCON, Submission to the CAESAR Competition," accessed: 2022-09-18. [Online]. Available: https://ascon.iaik.tugraz.at

[41] R. Maes, "Physically unclonable functions: Constructions, properties and applications," Heidelberg, Germany: *Springer*, 2013.

[42] M. Kaveh, M. R. Mosavi, "A lightweight mutual authentication for smart grid neighborhood area network communications based on physically unclonable function," *IEEE Systems J.*, vol. 14, no. 3, pp. 4535–4544, Sep. 2020.

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

[44] M. Tanveer, *et al.*, "ARAP-SG: Anonymous and reliable authentication protocol for smart grids," *IEEE Access*, vol. 9, pp. 143366–143377, Oct. 2021.

[45] M. Tanveer, *et al.*, "REAS-TMIS: Resource-efficient authentication scheme for telecare medical information system," *IEEE Access*, vol. 10, pp. 23008–23021, Feb. 2022.