Provably Secure and Lightweight Authentication and Key Agreement Protocol for Fog-Based Vehicular Ad-Hoc Networks

Syed Muhammad Awais , Wu Yucheng , Khalid Mahmood , *Senior Member, IEEE*, Mohammed J. F. Alenazi , *Senior Member, IEEE*, Ali Kashif Bashir , *Senior Member, IEEE*, Ashok Kumar Das , *Senior Member, IEEE*, and Pascal Lorenz , *Senior Member, IEEE*

***Abstract*— The increase in popularity of vehicles encourages the development of smart cities. With this advancement, vehicular ad-hoc networks, or VANETs, are now frequently utilized for inter-vehicular communication to gather data regarding traffic congestion, vehicle location, speed, and road conditions. Such a public network is open to various security risks. Overall, protecting personal information on VANET is a vital responsibility. The integration of fog computing and VANETs has gained significant importance in recent years, driven by advancements in cloud computing, Internet of Things (IoT) technologies, and intelligent transportation systems. However, ensuring secure communication in fog-based VANETs remains a major challenge. To overcome this challenge, we introduce a novel authenticated key agreement protocol that achieves mutual authentication, generates a secure session key for secret communication, and provides privacy protection without the use of bilinear pairing. We rigorously prove the security of our proposed protocol, which is designed specifically for fog-based VANETs, and has been shown to meet their stringent security requirements. Moreover, we performed formal and informal analysis that shows our proposed protocol is highly efficient,our protocol's computational and communication overhead are lower than those of other relevant protocols by 45.570% and 29.432%, respectively. Finally we use NS-3 simulation to prove that our proposed algorithm is a practical and scalable solution for secure communication in fog-based VANETs.**

***Index Terms*— Fog computing, authentication, key agreement, privacy, VANET, security.**

I. Introduction

The Internet of Things (IoT) is poised to become the next major productivity driver after the computer and the Internet. Its ability to remotely monitor and diagnose through various sensing technologies and communication modes has led to applications in Intelligent Transportation Systems (ITS), smart housing, and environmental protection. Zhang et al. [1] show that IoT can reduce traffic congestion, improve transport efficiency, and minimize environmental pollution. However, as more devices connect to ITS, increased computational and processing costs pose significant challenges [2], [3], [4], [5].

Fog nodes in traffic control systems can help address traffic congestion. These nodes, an extension of VANETs, facilitate efficient road resource allocation through enhanced communication and processing. Onboard units (OBUs) in vehicles support vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications [6], [7], [8]. V2V allows vehicles to dynamically adjust their routes based on traffic conditions, while V2I enables vehicles to request services through roadside units (RSUs), which relay messages to service providers or authentication centers. Dedicated short-range communication (DSRC) in VANETs facilitates real-time data transmission.

Fog computing plays a crucial role in VANETs, enabling real-time traffic monitoring and control. It brings several benefits:

1) Risky Driving Detection: Sensors on cars and roadsides detect risky driving practices, issue early warnings, and impose sanctions if necessary [9].

2) Traffic Control: Real-time traffic control adjusts signals according to road conditions and vehicle speeds, reducing delays on clear roads.

3) Emergency Response: Real-time reminders adjust traffic lights when emergency vehicles approach by leveraging sensor data.

Fog computing, therefore, enhances VANETs by providing efficient traffic management and real-time response capabilities. However, integrating fog computing into VANETs

Manuscript received 7 January 2024; revised 8 May 2024 and 29 June 2024; accepted 29 August 2024. This work was supported in part by King Saud University, Riyadh, Saudi Arabia, under Project RSPD2024R582. The Associate Editor for this article was S. H. Ahmed Shah. *(Corresponding authors: Pascal Lorenz; Khalid Mahmood.)*

Syed Muhammad Awais and Wu Yucheng are with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China (e-mail: syedmowais87@gmail.com; wuyucheng@cqu.edu.cn).

Khalid Mahmood is with the Graduate School of Intelligent Data Science, National Yunlin University of Science and Technology, Douliu, Yunlin 64002, Taiwan (e-mail: khalidm.research@gmail.com).

Mohammed J. F. Alenazi is with the College of Computer and Information Sciences, Department of Computer Engineering, King Saud University, Riyadh 12372, Saudi Arabia (e-mail: mjalenazi@ksu.edu.sa).

Ali Kashif Bashir is with the Department of Computing and Mathematics, Manchester Metropolitan University, M15 6BH Manchester, U.K., also with the Department of Computer Science and Mathematics, Lebanese American University, Beirut 03797751, Lebanon, and also with Centre for Research Impact and Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, Punjab (e-mail: dr.alikashif.b@ieee.org).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India, and also with the Department of Computer Science and Engineering, College of Informatics, Korea University, Seongbuk-gu, Seoul 02841, South Korea (e-mail: ashok.das@iiit.ac.in).

Pascal Lorenz is with the IRIMAS, University of Haute Alsace, 68100 Mulhouse, France (e-mail: lorenz@ieee.org).

TABLE I

Summary of Existing Authentication Schemes for VANETs

Protocol	Year	Cryptographic Primitives	Benefits	Drawbacks/Flaws
Jia *et al.* [10]	2019	Signature based Bilinear pairing-based AKA	* Session key exchange	— Violates anonymity — Computationally expensive
Wazid *et al.* [13]	2019	Public-key cryptography	* Secure key exchange * Supports efficient key distribution	— High computation and communication costs
Zhang *et al.* [19]	2020	ECC	* authentication efficiency * Supports efficient key distribution	— Management issues, Network delays
Wang *et al.* [15]	2020	Use self-certified public key cryptography	* Resistance against insider attack	— Batch verification is not considered
Chen *et al.* [23]	2021	Public-key cryptography	* Provides mutual authentication	— Lacks analysis of resilience to attacks — No support for key revocation or renewal in case of a security breach
Li *et al.* [14]	2022	Symmetric-key cryptography	* Provides privacy protection for user identities	— Lacks analysis of resilience to attacks
Salem *et al.* [20]	2023	Symmetric key-based authentication	* Resistant against semi-trusted third party	— High Computation Cost
Ma *et al.* [21]	2023	Certificateless Identity Authentication	* Achieves secure and efficient key agreement * Resistant to attacks against identity verification	— Limited scalability

requires secure communication protocols. Authenticated Key Agreement (AKA) protocols can address these challenges by achieving mutual authentication between entities and generating a common session key. Unfortunately, many existing AKA protocols are unsuitable for fog environments due to the computationally expensive bilinear pairings, which are impractical for real-time, high-speed applications like VANETs.

Jia et al. [10] presented a signature-based protocol with bilinear pairing-based AKA for mutual authentication and session key agreement, but it violates anonymity and is computationally demanding. Ismail et al. [11] proposed a group signatures-based protocol that compromises anonymity due to the trade-off between group size and anonymity level. Wei et al. [12] introduced a lightweight, conditionally privacy-preserving AKA scheme for fog-based VANETs using symmetric cryptography, which significantly reduces computational and communication overhead. Wazid et al. [13] developed a secure key management and user authentication scheme for fog computing with a hierarchical structure and centralized key server using elliptic curve cryptography (ECC). Li et al. [14] proposed a lightweight mutual authentication protocol for fog-enabled social IoV with secure hashing and symmetric encryption to protect against attacks. However, Wang et al. [15] proposed a multi-server authentication scheme that lacks batch verification and suffers from replay attacks.

Zhao et al. [16] introduced the Conditional Choice Probability Federated Deep Learning Algorithm to enhance distributed learning. Ibrar et al. [17] proposed an SDN-enabled adaptive clustering technique to address rapid VANET fluctuations. Zheng [18] developed a digital twin-based social relationship model for managing and analyzing vehicle interactions. Zang et al. [19] introduced a solution for fog computing that reduces server-side response time and improves authentication efficiency. However, network delays still occur due to the volume of vehicles. Salem et al. [20] proposed a symmetric-key authentication protocol that suffers from high computation costs. Ma et al. [21] presented a certificateless identity protocol with limited scalability. Chen et al. [22] described a confidential computing-based key transfer protocol for VANETs to secure vehicular communication. These studies highlight advancements and challenges in SIoV, providing a foundation for future research.

In summary, Table I compares various VANET authentication schemes based on cryptographic primitives, benefits, and drawbacks. Despite several proposed protocols, many suffer from security threats. The new AKA methodology presented here improves efficiency by excluding bilinear pairings.

A. Motivation and Contributions

Integrating fog computing with VANETs requires secure communication protocols. To meet these demands, we've developed a novel authenticated key agreement mechanism for fog-based VANETs that ensures mutual authentication, secure session key generation, and privacy protection without using bilinear pairings.

- We have designed a new four-party authentication protocol specifically for VANETs, utilizing ECC and lightweight cryptographic techniques such as concatenation, bit-wise XOR, and hash functions. This protocol enhances the security and efficiency of communications within the network.
- We propose a unique authentication key agreement protocol that facilitates secure interactions in fog-based VANETs. This protocol operates without the need for bilinear pairing, simplifying the cryptographic process while maintaining robust security.
- To further bolster the security of our protocol, we have incorporated biometric-based authentication for users. This addition significantly enhances the security features of our system, providing a more reliable and secure user verification method.
- The security of our proposed protocol has been rigorously tested both formally and informally, using the ROM. It has been proven to meet the stringent security requirements essential for fog-based VANETs.
- We have conducted a comprehensive performance analysis of our protocol. The results show that our protocol significantly reduces both computational and communication overhead by 45.570% and 29.432%, respectively, compared to other relevant protocols.

B. Paper Organization

The structure of this paper is as follows: Section II introduces the system model and necessary preliminaries.

Section III details the proposed key agreement protocol. Section IV conducts a security analysis, and Section V evaluates the protocol's performance. Section VI presents NS3 simulations. The paper concludes in Section VII with a summary of contributions and future work directions.

II. Preliminaries

The proposed key agreement protocol in this paper is based on elliptic curve cryptography (ECC) and assumes a cyclic additive group G with q-order, where q is a large prime number. A generator point $P \in G$ is used for computations. The paper makes use of several assumptions, including the ECDL assumption, which is the problem of solving for an unknown $x \in \mathbb{Z}_q$ given two points P and xP in G. The ECCDH assumption is also used, which involves calculating an unknown xyP in G given three points P, xP, and yP. Additionally, the ECDDH assumption is utilized, which is the difficulty of deciding whether four points P, xP, yP, and zP in G satisfy $Z = xyP$ for unknown numbers x, y, and z in $\mathbb{Z}_q$. These assumptions provide the foundation for the proposed key agreement protocol and are discussed in detail in Section III.

A. Adversarial Model

We utilize the threat models proposed by Dolev and Yao [24] as well as Canetti and Krawczyk [25], commonly referred to as the DY and CK threat models, respectively, to delineate the capabilities of the adversary $\mathcal{A}$. The DY model delineates the adversary's ability to access the communication channel, while the CK model extends the capabilities of the DY model by allowing the adversary to compromise session key information. Consequently, we consider the adversary to possess the following capabilities:

- $\mathcal{A}$ has access to the public channel.
- $\mathcal{A}$ can intercept public messages, modify them, store them, or replay them.
- $\mathcal{A}$ has complete access to the public identities of all participants.
- $\mathcal{A}$ can physically access the vehicle's on-board memory unit and extract stored information from it.
- $\mathcal{A}$ is capable of conducting enumeration and guessing attacks.

B. Hash Function

The function $h : \{0,1\}^* \rightarrow \{0,1\}^{las}$ is a pseudo-random hash function. Where $h : \{0,1\}^*$ denoted the set of all possible binary string of any lenght, and $\{0,1\}^{las}$ denotes the set of binary strings with a fixed length of las bits. The advantage of an adversary $\mathcal{A}$'s in discovering a collision during hash execution is represented by $Advt_A$. The advantage $Advt_A$ is calculated as the probability that the adversary $\mathcal{A}$ can successfully find a collision given two randomly chosen inputs n_1 and n_2, where $n_1 \neq n_2$ and the hash of both inputs $h(n1) = h(n2)]$ are equal [26]. In mathematical terms, $Advt_A$ is expressed as:$Advt_A = Pr[(n_1, n_2) \in A : n_1 \neq n_2, h(n1) = h(n2)]$.

C. Security Requirements

Ensuring confidentiality is a critical aspect of fog-based VANETs, and any AKA protocol for such environments must satisfy several security requirements. These include the following:

1) Mutual Authentication: The AKA protocol should support mutual authentication to ensure the validity of all participants.
2) Session Key Agreement: A common session key must be generated among the participants to encrypt messages and protect the confidentiality of future interactions.
3) Untraceability and Anonymity: The protocol must prevent adversaries from tracking a user's behavior and identity, even if they intercept messages during transmission.
4) Perfect Forward Secrecy: The AKA protocol should provide perfect forward secrecy to protect the confidentiality of messages exchanged during previous interactions. This means that even if an attacker gains access to the participants' long-term private keys, the session key established during the previous session should still be safe.
5) Resistance to Stolen Verifier Attack: Even if an attacker steals the verifies table contents from CS, they should be unable to deduce users' private keys.
6) Session key stolen: The session key stolen attack is when an A intercepts and steals the session key used to encrypt data being transmitted between communicating parties, allowing them to decrypt and potentially modify the data
7) Man-in-the-Middle Attack: The attacker should be unable to act as legitimate user and deceive CS.

D. System Model

Fig. 1 illustrates the proposed protocol's system paradigm for Fog-based VANETs. The protocol involves the following participants: a vehicle user U_i, a Roadside Unit RSU_k, a fog node FN_j, and a Cloud Server CS.

- The protocol for fog-based VANETs involves several participants, includingU_i - the ith vehicle user, who controls their vehicle equipped with an OBU that communicates with FN_j through a nearby RSU_k. To gain access to the system, U_i must obtain a smart card from the CS and authenticate the process with correct password and smart card information.
- An RSU_k is a wireless communication device used only as a gateway in VANETs that is mounted on roadside infrastructure. Its primary function is to broadcast and relay messages between vehicles and other network entities.
- FN_j is an untrusted participant in the protocol, with its own computing and storage capacity. Its primary responsibility is to facilitate the exchange messages of authentication between CS and U_i. Once the authentication process is complete, FN_j collaborates with CS and U_i to establish a common session key.
- CS is a trustworthy cloud service provider responsible for generating the system's public parameters and the

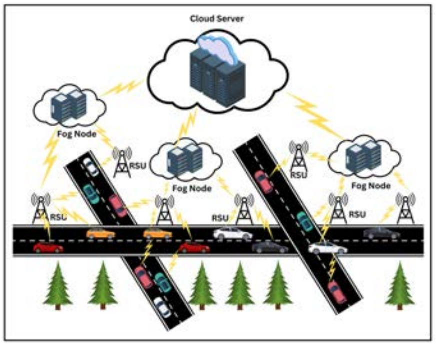

Fig. 1. Illustration of fog-based architecture for VANET.

TABLE II
NOTATIONS

$Notation$	$Description$
r_i	Set of Random numbers
U_i, RSU_k, FN_j	Vehicle user , Roadside unit and Fog node
CS	Cloud Server of the system
ID_{U_i}	U_i Identity
$D_{ID_i}, D_{RSU_k},$	Secret keys of U_i , RSU_k
$D_{ID_j}, SK_{i-k-j-cs}$	Secret keys of FN_j and CS
s	Secret key of CS
SC	Smart card
$\mathcal{A}$	Adversary
$h(.)$	Secure one-way hash function
$\oplus$	Bitwise XOR operation
$\|$	Concatenation operation

private keys. Additionally, CS offers registration services for U_i, RSU_k and FN_j, ensuring that each participant is registered according to their respective requests. To facilitate subsequent authentication, CS stores verifiers derived from (U_i, RSU_k, FN_j) identities and the primary keys.

III. Proposed Protocol

In this section, we introduce a novel three-party authentication key agreement (AKA) mechanism for fog-based VANETs without bilinear pairings. The list of various notations with their significance is provided in Table II. The following steps provide a detailed breakdown of our process.

A. Initialization Phase

In this phase, CSÂ creates the necessary parameters and security parameter k, CS also configures the system to carry out further operations such as vehicle,roadside unit and fog node registration.

Step-1: CS selects a generator P and q-order additive group G as an elliptic curve point (a formal definition is provided in Section II).

Step-2: Randomly choosing $s \in Z_q^*$ and q, CS determines $P_{pub} = sP$.

Step-3: CS selects secure cryptographic hash functions $h(.) : \{0,1\}^* \to \{0,1\}^{ln}$.

Step-4: Lastly, CS publishes system parameters prms= $\{k, q, P, G, P_{pub}$,$h_i\}$ while keeping s a secret.

B. Vehicle User Registration Phase

To be recognized as an authorized participant in the system and receive the associated parameters, the vehicle user U_i registers itself with CS must complete the following process:

Step-1: U_i selects its identity ID_{U_i} and then imprint BIO_{u_i}, and computes $GenBIO_{U_i} = (\sigma_{U_i}, \delta_{U_i})$ and transmit ID_{U_i} to CS.

Step-2: On receiving ID_{U_i}, CS calculates $D_{ID_i} = h(s\|ID_{U_i})$, then generate: $GenBIO_{U_i}$.

Step-3: In the end, the credentials ID_{U_i}, $GenBIO_{U_i}$ and D_{ID_i} are securely stored on smart card, CS hands over the smart card to U_i.

C. Roadside Unit Registration Phase

Likewise to vehicle user U_i, the roadside unit RSU_k also undergoes the registration process with CS with the following steps:

Step-1: RSU_k transmit its identification ID_{RSU_k} to CS.

Step-2: On arrival of ID_{RSU_k}, CS calculates $D_{ID_k} = h(s\|ID_{RSU_k})$.

Step-3: Finally, the credentials ID_{RSU_k} and D_{ID_k} are securely stored on smart card, CS hands over the smart card to RSU_k.

D. Fog Node Registration Phase

Similarly to RSU_k, the FN_j also undergoes the registration process with CS with the following steps:

Step-1: FN_j transmits its ID_{FN_j} identity to CS.

Step-2: After receiving ID_{FN_j}, CS calculates $D_{ID_j} = h(s\|ID_{FN_j})$. Once calculated, D_{ID_j} is then securely transmitted back to CS via secure channel.

Step-3: With the secure receipt and storage of both ID_{FN_j} and D_{ID_j}, FN_j successfully completes the registration process.

E. Mutual Authentication and Key Agreement Phase

To implement mutual authentication and generate a shared session key, the cloud server CS, the fog node FN_j, the roadside unit RSU_k and the vehicle user U_i each individually complete the actions listed below. The interaction between them is depicted in Fig. 2.

1) Firstly,U_i inputs his/her identity ID_{U_i} and also imprints his/her biometric BIO_{U_i} at the sensor of device. Then computes $\sigma' = Rep(BI\bar{O}'_{U_i}, \delta_{U_i})$. After that, a random number is selected $r_1 \leftarrow Z_q^*$, $R_1 = r_1 P$, $\bar{R}_1 = r_1 P_{pub}$ and computes TID_{U_i}=$ID_{U_i} \oplus \bar{R}_1$, $\alpha = h((ID_{U_i}\|TID_{u_i}\|\bar{R}_1\|D_{ID_i})$ and transmits message $(M_1 = TID_{U_i}, R_1, \alpha)$ to RSU_k via public channel.

2) After receiving the M_1 from U_i, RSU_k selects a random number, $r_2 \leftarrow Z_q^*, R_2 = r_2 P, \bar{R}_2 = r_2 P_{pub}$ and computes $TID_{RSU_k} = ID_{RSU_k} \oplus \bar{R}_2$ and $\beta = h(ID_{RSU_k}\|TID_{RSU_k}\|\bar{R}_1\|\bar{R}_2\|D_{ID_k})$. finally RSU_kforwards messages $(M_2 = M_1, TID_{RSU_k}, R_2, \beta)$to FN_j using a public channel.

3) After receiving message M_2 from RSU_k. FN_j selects a random number $r_3 \leftarrow Z_q^*, R_3 = r_3 P, \bar{R}_3 = r_3 P_{pub}$. Then computes, $TID_{FN_j} = ID_{FN_j} \oplus \bar{R}_3$ and $\gamma = h(ID_{FN_j}\|TID_{FN_j}\|\bar{R}_2\|\bar{R}_3\|D_{ID_j})$. Finally, FN_j transmits $(M_3 = M_2, TID_{FN_j}, R_3, \gamma)$ to CS via public channel.

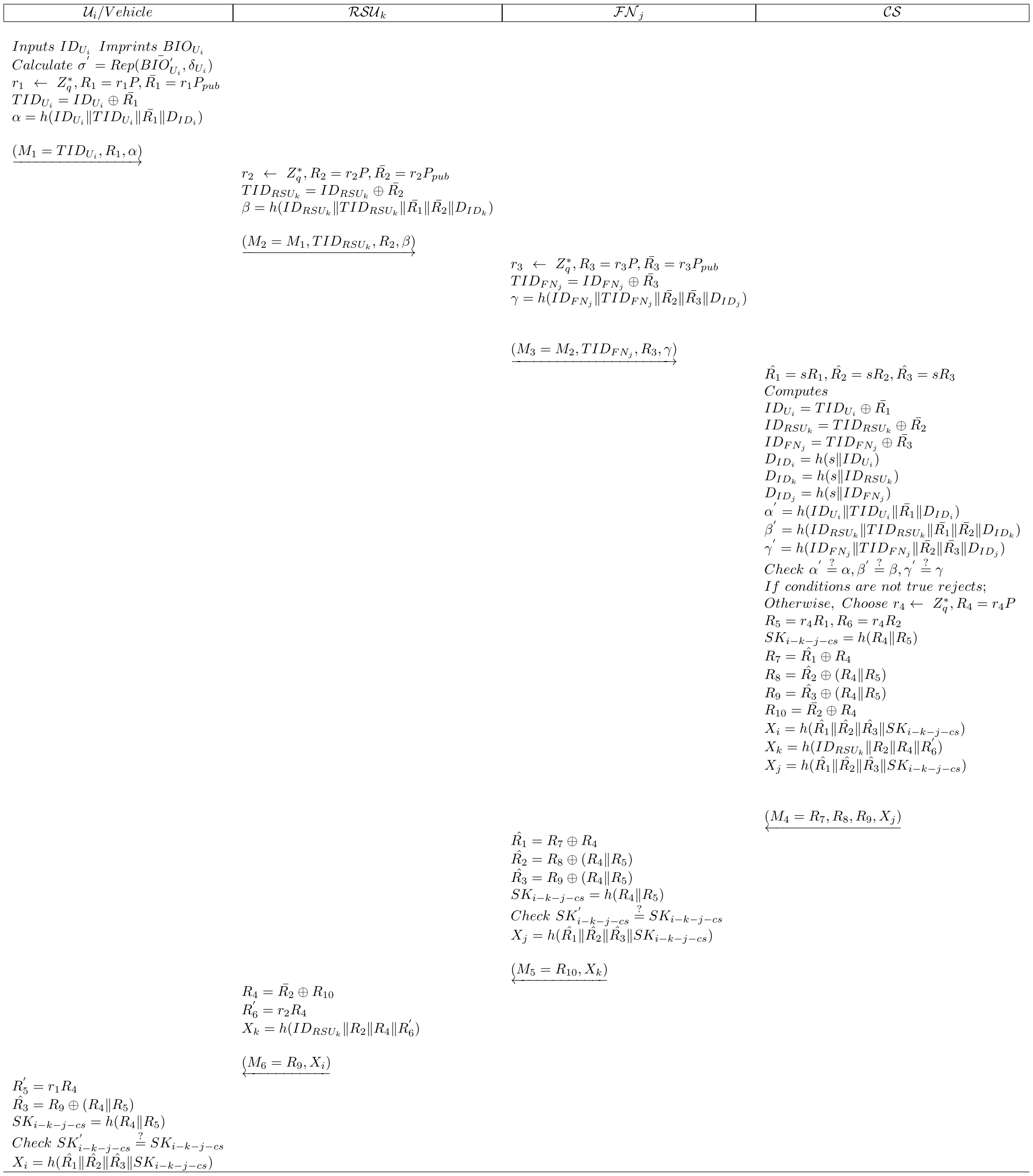
$\mathcal{U}_i/Vehicle$
$\mathcal{RSU}_k$
$\mathcal{FN}_j$
$\mathcal{CS}$
$Inputs\ ID_{U_i}\ \ Imprints\ BIO_{U_i}$
$Calculate\ \sigma' = Rep(B\bar{IO}'_{U_i}, \delta_{U_i})$
$r_1 \leftarrow Z_q^*, R_1 = r_1P, \bar{R}_1 = r_1P_{pub}$
$TID_{U_i} = ID_{U_i} \oplus \bar{R}_1$
$\alpha = h(ID_{U_i} \| TID_{U_i} \| \bar{R}_1 \| D_{ID_i})$
$\xrightarrow{(M_1 = TID_{U_i}, R_1, \alpha)}$
$r_2 \leftarrow Z_q^*, R_2 = r_2P, \bar{R}_2 = r_2P_{pub}$
$TID_{RSU_k} = ID_{RSU_k} \oplus \bar{R}_2$
$\beta = h(ID_{RSU_k} \| TID_{RSU_k} \| \bar{R}_1 \| \bar{R}_2 \| D_{ID_k})$
$\xrightarrow{(M_2 = M_1, TID_{RSU_k}, R_2, \beta)}$
$r_3 \leftarrow Z_q^*, R_3 = r_3P, \bar{R}_3 = r_3P_{pub}$
$TID_{FN_j} = ID_{FN_j} \oplus \bar{R}_3$
$\gamma = h(ID_{FN_j} \| TID_{FN_j} \| \bar{R}_2 \| \bar{R}_3 \| D_{ID_j})$
$\xrightarrow{(M_3 = M_2, TID_{FN_j}, R_3, \gamma)}$
$\hat{R}_1 = sR_1, \hat{R}_2 = sR_2, \hat{R}_3 = sR_3$
$Computes$
$ID_{U_i} = TID_{U_i} \oplus \bar{R}_1$
$ID_{RSU_k} = TID_{RSU_k} \oplus \bar{R}_2$
$ID_{FN_j} = TID_{FN_j} \oplus \bar{R}_3$
$D_{ID_i} = h(s \| ID_{U_i})$
$D_{ID_k} = h(s \| ID_{RSU_k})$
$D_{ID_j} = h(s \| ID_{FN_j})$
$\alpha' = h(ID_{U_i} \| TID_{U_i} \| \bar{R}_1 \| D_{ID_i})$
$\beta' = h(ID_{RSU_k} \| TID_{RSU_k} \| \bar{R}_1 \| \bar{R}_2 \| D_{ID_k})$
$\gamma' = h(ID_{FN_j} \| TID_{FN_j} \| \bar{R}_2 \| \bar{R}_3 \| D_{ID_j})$
$Check\ \alpha' \overset{?}{=} \alpha, \beta' \overset{?}{=} \beta, \gamma' \overset{?}{=} \gamma$
$If\ conditions\ are\ not\ true\ rejects;$
$Otherwise,\ Choose\ r_4 \leftarrow Z_q^*, R_4 = r_4P$
$R_5 = r_4R_1, R_6 = r_4R_2$
$SK_{i-k-j-cs} = h(R_4 \| R_5)$
$R_7 = \hat{R}_1 \oplus R_4$
$R_8 = \hat{R}_2 \oplus (R_4 \| R_5)$
$R_9 = \hat{R}_3 \oplus (R_4 \| R_5)$
$R_{10} = \bar{R}_2 \oplus R_4$
$X_i = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$
$X_k = h(ID_{RSU_k} \| R_2 \| R_4 \| R'_6)$
$X_j = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$
$\xleftarrow{(M_4 = R_7, R_8, R_9, X_j)}$
$\hat{R}_1 = R_7 \oplus R_4$
$\hat{R}_2 = R_8 \oplus (R_4 \| R_5)$
$\hat{R}_3 = R_9 \oplus (R_4 \| R_5)$
$SK_{i-k-j-cs} = h(R_4 \| R_5)$
$Check\ SK'_{i-k-j-cs} \overset{?}{=} SK_{i-k-j-cs}$
$X_j = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$
$\xleftarrow{(M_5 = R_{10}, X_k)}$
$R_4 = \bar{R}_2 \oplus R_{10}$
$R'_6 = r_2R_4$
$X_k = h(ID_{RSU_k} \| R_2 \| R_4 \| R'_6)$
$\xleftarrow{(M_6 = R_9, X_i)}$
$R'_5 = r_1R_4$
$\hat{R}_3 = R_9 \oplus (R_4 \| R_5)$
$SK_{i-k-j-cs} = h(R_4 \| R_5)$
$Check\ SK'_{i-k-j-cs} \overset{?}{=} SK_{i-k-j-cs}$
$X_i = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$

Fig. 2. Fog-Based Authentication and Key Agreement Protocol for VANETs.

4) After receiving the message M_3 from FN_j, CS calculates $\hat{R}_1 = sR_1$, $\hat{R}_2 = sR_2$, $\hat{R}_3 = sR_3$. Then it computes $ID_{U_i} = TID_{U_i}\_ \oplus \bar{R}_1$, $ID_{RSU_k} = TID_{RSU_k} \oplus \bar{R}_2$, $ID_{FN_j} = TID_{FN_j} \oplus \bar{R}_3$, $D_{ID_i} = h(s \| ID_{U_i})$, $D_{ID_k} = h(s \| ID_{RSU_k})$, $D_{ID_j} = h(s \| ID_{FN_j})$, $\alpha' = h(ID_{U_i} \| TID_{U_i} \| \bar{R}_1 \| D_{ID_i})$, $\beta' = h(ID_{RSU_k} \| TID_{RSU_k} \| \bar{R}_1 \| \bar{R}_2 \| D_{ID_k})$, $\gamma' = h(ID_{FN_j} \| TID_{FN_j} \| \bar{R}_2 \| \bar{R}_3 \| D_{ID_j})$, then verifies if the $\alpha' \overset{?}{=} \alpha, \beta' \overset{?}{=} \beta, \gamma' \overset{?}{=} \gamma$. If the conditions are not true then CS rejects. Otherwise CS chooses a random number $r_4 \leftarrow Z_q^*, R_4 = r_4P_{pub}$ and calculate $R_5 = r_4R_1$, $R_6 = r_4R_2$ and CS selects a session key $SK_{i-k-j-cs} = h(R_4 \| R_5)$, then computes $R_7 = \hat{R}_1 \oplus R_4$, $R_8 = \hat{R}_2 \oplus (R_4 \| R_5)$, $R_9 = \hat{R}_3 \oplus (R_4 \| R_5)$, $R_{10} = \bar{R}_2 \oplus R_4$, $X_i = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$, $X_{kf} = h(ID_{RSU_k} \| R_2 \| R_4 \| R'_6)$, $X_j = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$. Finally CS transmits $(M_4 = R_7, R_8, R_9, X_j)$ to FN_j via insecure channel.

5) FN_j receives message M_4 and determines $\hat{R}_1 = R_7 \oplus R_4$, $\hat{R}_2 = R_8 \oplus (R_4 \| R_5)$, $\hat{R}_3 = R_9 \oplus (R_4 \| R_5)$. Then computes $SK_{i-k-j-cs} = h(R_4 \| R_5)$, and verifies $SK'_{i-k-j-cs} \overset{?}{=} SK_{i-k-j-cs}$ if it is not verified, FN_j terminate the session. Otherwise, FN_j calculate $X_j = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$. Finally FN_j transmits $(M_5 = R_{10}, X_k)$ to RSU_k via public channel.

6) RSU_k receives message M_5 and determines $R_4 = \bar{R}_2 \oplus R_{10}$, $R'_6 = r_2R_4$ and calculates $X_k = h(ID_{RSU_k} \| R_2 \| R_4 \| R'_6)$. Finally RSU_k transmits $(M_6 = R_9, X_i)$ to U_i via insecure channel.

7) U_i receives message M_6 and determines $R'_5 = r_1R_4$, $\hat{R}_3 = R_9 \oplus (R_4 \| R_5)$. Then computes $SK_{i-k-j-cs} = h(R_4 \| R_5)$, and verifies $SK'_{i-k-j-cs} \overset{?}{=} SK_{i-k-j-cs}$ if it is not verified, U_i terminate the session. Otherwise, U_i calculate $X_i = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$.

IV. Security Analysis

The formal and informal security analysis of the suggested protocol are described in this section.

A. Formal Security Analysis

This section first describes the security paradigm of our protocol. The security model discussed before is then used to demonstrate that devised protocol is secure. Our techniques pass the security requirements. The devised protocol's formal and informal evaluations of security are covered below.

B. Security Model

Our devised protocol security concept is based on the work of Bellare et al. [27], is then developed via a series of games between challenger C and adversary $\mathcal{A}$. Assume that I represents the participant's i-th instance, $\Lambda \in$(U,FN,CS) is represented by $\prod_{\Lambda}^{i}$ and $\sum$ stands for the devised protocol. In this model, A can ponder a number of oracle-related inquiries, and C will reply as displayed below.

- Send($\prod_{\Lambda}^{i}, m$): If A sends message m containing the inquiry, C runs the designed protocol in accordance with specified steps and reports the findings.
- Reveal($\prod_{\Lambda}^{i}$): In response to A's inquire, if $\prod_{\Lambda}^{i}$ is approved, C generates the session key. else, C returns $\perp$.
- Corrupt(ID_{U_i}): The forward security inquiry is simulated here. U_i's private key is returned by C if A issues the query with the identification ID_{U_i} for U_i.
- Execute($\prod_{U}^{i}, \prod_{RSU}^{k}, \prod_{FN}^{j}, \prod_{CS}^{k}$): This command imitate the adversary's $\mathcal{A}$ execution. a technique for reading or listening. When the protocol is being used, C displays all messages sent by instances ($\prod_{U}^{i}, \prod_{RSU}^{k}, \prod_{FN}^{j}, \prod_{CS}^{k}$).
- Test($\prod_{\Lambda}^{i}$): In the event that adversary $\mathcal{A}$ poses the question, C will randomly select $a, b \in \{0, 1\}$. If $b = 1$, A participates in $\prod_{\Lambda}^{i}$ by means of the session key. If not, C picks a random integer that matches the length of the session key and sends it to A.
  In the event that adversary $\mathcal{A}$ poses the inquiry, C will arbitrarily select $ab \in \{0, 1\}$. If $b = 1$, A participates in $\prod_{\Lambda}^{i}$ using the session key. Otherwise, C picks a number at random whose length is equal to the session key and transmits it to A.

Definition 1 (Partnership): We say that $\prod_{\Lambda}^{i}$ and $\prod_{\bar{\Lambda}}^{i}$ are partners if the instances $\prod_{\Lambda}^{i}$ and $\prod_{\bar{\Lambda}}^{i}$ possess the following characteristics.
1) $\prod_{\Lambda}^{i}$ can communicate with $\prod_{\bar{\Lambda}}^{i}$ directly and share information.
2) $\prod_{\Lambda}^{i}$ and $\prod_{\bar{\Lambda}}^{i}$ both use the same session key, SK.
3) Instances other than $\prod_{\Lambda}^{i}$ and $\prod_{\bar{\Lambda}}^{i}$ do not accept SK.

Definition 2 (Freshness): $\prod_{\Lambda}^{i}$ is considered fresh if it satisfies the following criteria.
1) The session key, SK, has been accepted by $\prod_{\Lambda}^{i}$.
2)No participant was questioned about falsified queries prior to acceptance.
3) $\prod_{\Lambda}^{i}$ don't or its partners have any queries to Reveal.

Definition 3 (Freshness of Session Key): In the case where $\prod_{\Lambda}^{i}$ and $\prod_{\bar{\Lambda}}^{i}$ are partners, with SK serving as a session key they share, SK is only valid if and only if both $\prod_{\Lambda}^{i}$ and $\prod_{\bar{\Lambda}}^{i}$ are valid.

Definition 4 (Advantage of Adversary): Assume that success (A) refers to the circumstance in which A executes a test ($\prod_{\Lambda}^{i}$) query using a new instance of $\prod_{\Lambda}^{i}$ and successfully returns the value b. The opponent that A is using to attack the AKA protocol $\sum$ is stated as

$$Adv_{\sum}^{AKA} =\mid 2Pr[Success(A)] - 1 \mid . \tag{1}$$

Definition 5 (AKA-secure): The protocol is considered to be AKA-secure if $Adv_{\sum}^{AKA}(A)$ is a negligible function for any polynomial adversary $\mathcal{A}$.

C. Provable Security

Using the security model outlined in Section V-A, we demonstrate in this part that the protocol we developed is secure for both authentication and key agreement.

Theorem 1: The protocol provided in Section IV cannot be compromised by any polynomial adversary $\mathcal{A}$ with a non-zero probability.

Proof 1: The detailed proof of this theorem is provided in the supplementary material.

D. Informal Security Analysis

In this section, we demonstrate that our devised protocol satisfies the security criteria outlined in Section II-B.

1) Mutual Authentication: No polynomial adversary can effectively forge a legal login or response message, as shown by the Theorem 1 proof. Therefore, by confirming whether or not their received information is legitimate, the participants might authenticate one another. Therefore, the devised protocol guarantees mutual authentication.

2) User Impersonation Attack: In order to carry out a successful U_i impersonation attack on our protocol, A would need to fabricate a legitimate login request message, M_1, containing the elements $(TID_{U_i}, \bar{R}_1, \alpha)$. However, accomplishing this task would require A to possess U_i's confidential credentials $(i.e., ID_{U_i}\ and\ BIO_{U_i})$, which are not available to A. Consequently, without access to these parameters, it is practically impossible for A to impersonate U_i. This underscores the protocol's effectiveness in preventing vehicle impersonation attacks.

3) Cloud Server Impersonation Attack: Let's consider A's role as a legitimate CS aiming to manipulate other entities. In this scenario, A can intercept M_3 and produce $M_4 = (R_6, R_7, R_8, R_9, R_{10}, X_i, X_j, X_k)$. It's essential to note that the creation of $M3$ relies solely on $CS's$ secret key, s, which A does not possess. Hence, without access to s, A cannot generate M_4. Consequently, this defense mechanism fortifies our protocol against cloud server impersonation attacks.

4) Session Key Agreement: All participants must be able to calculate the same value, $K = r_1, r_2, r_3, r_4$, and the shared session key, $SK_{i-k-j-cs} = h(R_4 \| R_5)$,in accordance with the instructions provided in Section $IV - D$. As a result, the AKA protocol can establish session key agreements.

5) User Anonymity: The identity ID_{U_i} of the vehicle user U_i is concealed by the protocol described in Section IV-D

in $TID_{U_i} = ID_{U_i} \oplus \bar{R}_1$). To find ID_{U_i} from TID_{U_i} the adversary must calculate $\bar{R}_1 = r_1 P_{pub}$ from $R_1 = r_1 P$, Despite the fact that the ECDL problem cannot be solved, our devised protocol ensures user anonymity.

6) Untraceability: The participants, like U_i, RSU_k, FN_j, and CS, select the random integers r_1, r_2, r_3, r_4 to compute $R_1 = r_1 P$, $R_2 = r_2 P$, $R_3 = r_3 P$, $R_4 = r_4 P$. As a result, the participants' actions cannot be tracked by the attacker. So, our protocol support untraceability.

7) Perfect Forward Secrecy: Consider a scenario in which attacker obtains the smart card and intercepts the messages $R_1 = r_1 P$, $R_2 = r_2 P$, $R_3 = r_3 P$, $R_4 = r_4 P$. The adversary must calculate $K = r_1, r_2, r_3, r_4 P$, or solve the $ECCDH$ problem, to obtain the value of $SK_{i-k-j-cs} = h(R_4 \| R_5)$. Due to the intractable nature of the $ECCDH$ assumption, devised protocol offer complete forward secrecy.

8) Known Session Key Attack: The session keys $SK = h(K, R_1, R_2, R_3, R_4)$ $K = r_1 P, r_2 P, r_3 P, r_4 P$ used in the protocol discussed in Section IV-D are distinct for each session since r_1, r_2, r_3, r_4 are random values. As a result, the confidentiality of other session keys is unaffected even if one is revealed.

9) Man-in-the-Middle Attack: Assume A is aware of ID_{U_i}, ID_{RSU_k} and ID_{FN_j}. A wants to create valid messages (M_1, M_2, M_3, M_4, M_5, M_6). To create a fake M_1, A randomly chooses the number $\bar{r}_1$, computes $R_1 = \underline{r_1 P}$, $\bar{R}_1 = r_1 P_{pub}$, and then calculates $TID'_{U_i} = ID_{U_i} \oplus \bar{R}_1$). Without D_{ID_i}, it is challenging for A to determine the value. Similar to this, A finds it difficult to forge without D_{ID_k} and D_{ID_j}. So, A is unable to compute M_2 and M_3. Additionally, without $CS's$ private key, A is unable to produce M_4, M_5 and M_6. Our protocol would therefore be able to prevent from attackers in the middle.

10) Ephemeral leakage attack: Assuming that an A can deduce the ephemeral secrets of any particular session; A would still need to rebuild the session key. In our proposed protocol, the ephemeral secrets R_4 and R_5 are crucial for creating the session key. Notably, during the authentication phase, R_4 and R_5 are not transmitted over a public channel nor exposed to potential interception, as they are protected by the irreversible characteristic of the hash function. Consequently, A can't determine a session key, thereby affirming that our protocol is robust against an ephemeral leakage attack.

11)Resistance to Stolen Verifier Attack: Instead of saving the verifier table, participants in the devised protocol discussed in Sec-IV need their private key, which is required to authenticate with other participants. Consequently, the suggested approach is resistant to a stolen verifier attack.

V. Performance Analysis

In this section compares the effectiveness of devised protocol with similar protocols from [28], [29], [30], [31], and [32]. We've introduced a four-party authentication system in our proposed protocol, and no prior protocols address this concept in existing literature. Consequently, we will evaluate the overhead comparison for three entities U_i, FN_j, and CS between our proposed protocol and the existing literature. Our protocol outperforms similar protocols regarding security characteristics, communication, and computation overhead. The following parts provide a complete comparison and experiment:

TABLE III

Cryptographic Operations With Their Execution Time (In Milliseconds)

Function	*Execution Time*		
	Arduino	Low End Machine	High-end Machine
T_{owh}	1.930 ms	0.0812 ms	0.0083 ms
$T_{enc/dec}$	0.890 ms	0.0662 ms	0.0059 ms
T_{pm}	0.732 ms	0.0574 ms	0.0032 ms
T_{pa}	0.539 ms	0.0312 ms	0.0012 ms
T_{bp}	2.103 ms	0.672 ms	0.031 ms

A. Experimental Setup

We executed the proposed protocol to measure the computational overhead through experimental results. Key cryptographic operations were carried out at both the vehicle end U_i and the FN_j end FN_j using Raspberry Pi and desktop systems respectively. The Raspberry Pi used in these experiments is equipped with a 1.4 GHz quad-core processor and 512 MB LPDDR2 SDRAM operating at a clock speed of 16 MHz. On the other hand, the desktop system specifications include a Linux OS, an Intel Core i5 CPU, and 32 GB RAM with a processing speed of 4.2 GHz, utilizing the PyCharm library. Details about these cryptographic operations, including their descriptions and execution times, are provided in Table III.

B. Computation Overhead

The total execution time of cryptographic operations constitutes the computing overhead in an authentication process. The following cryptographic functions and execution time of each cryptographic operation listed in Table III are considered when calculating the computational overhead of the proposed and related protocols:

- T_{owh}: One-way hash function
- $T_{enc/dec}$: Symmetric encryption and decryption
- T_{pm}: EC point multiplication
- T_{pa}: EC point addition
- T_{bp}: EC bilinear pairing

At the end of the computation U_i, FN_j and CS calculate the overhead in terms of computation. The comparison does not include the registration phase because it is a one-time operation. Therefore, while calculating the computing overhead, it is important to consider the login, authentication, and key agreement phases. In our protocol, U_i performs the hash function three times and two times point multiplication. The computation delay for U_i is therefore $(3 \times 0.0812) + (2 \times 0.0574) \approx 0.358$ ms. Similar to this, the hash function is run three times and two times point multiplication at FN_j and the computational delay at FN_j is therefore $(3 \times 1.930) + (2 \times 0.732) \approx 7.254$ ms. Whereas it is run nine times hash function and five times point multiplication at CS therefore the computational delay is $(9 \times 0.0083) + (5 \times 0.0032) \approx 0.091$ ms respectively. As a result, our protocol's overall computation overhead is 11.2453, and the computational overhead of our protocol is 49.095%, 53.641%,

TABLE IV

Computation Overheads Comparison

$Protocol$	U_i	FN_j	CS	$Total\ Overhead$
Proposed	$3T_{owh}$ + $2T_{pm}$ ≈ 0.358 ms	$3T_{owh}$ + $2T_{pm}$≈ 7.254 ms	$9T_{owh}$+$5T_{pm}$ ≈ 0.091 ms	7.703 ms
Tomar *et al.* [28]	$8T_{owh}$+$6T_{pm}$ ≈ 0.994 ms	$5T_{owh}$+$6T_{pm}$ ≈ 14.031 ms	$9T_{owh}$+$10T_{pm}$ ≈ 0.107 ms	15.132 ms
Cui *et al.* [29]	$8T_{owh}$+$3T_{pm}$ ≈ 0.821 ms	$7T_{owh}$+$3T_{pm}$ ≈ 15.706 ms	$10T_{owh}$ +$2T_{pm}$ ≈ 0.089 ms	16.616 ms
Jiang *et al.* [30]	$7T_{owh}$+$5T_{pm}$ ≈ 0.855 ms	$4T_{owh}$ ≈ 7.720 ms	$13T_{owh}$+$5T_{pm}$ ≈ 0.123 ms	8.699 ms
Chen *et al.* [31]	$9T_{owh}$+$4T_{pm}$+$1T_{pa}$ ≈ 0.992 ms	$8T_{owh}$+$7T_{pm}$+$3T_{pa}$ ≈ 22.181 ms	$9T_{owh}$+$8T_{pm}$+$2T_{pa}$ ≈ 0.103 ms	23.276 ms
Oh *et al.* [32]	$12T_{owh}$+$5T_{pm}$+$4T_{bp}$ ≈ 3.949 ms	$1T_{owh}$+$3T_{pm}$+$3T_{bp}$ ≈ 10.435 ms	$10T_{owh}$+$1T_{pm}$ ≈ 0.086 ms	14.470 ms

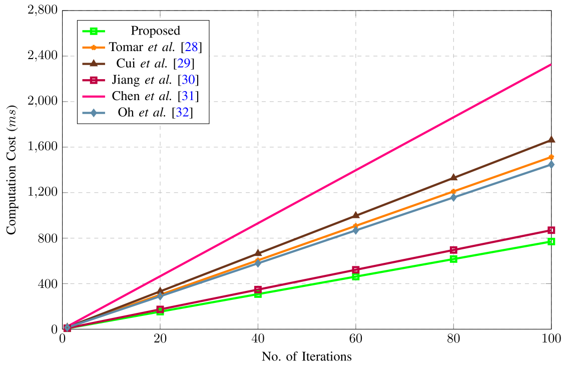

Fig. 3. Analysis of Computation Overheads.

11.449%, 66.906%, and 46.766% less than that of [28], [29], [30], [31], and [32] in that order. When compared to related protocols, our devised protocol reduces computing overhead by an average of 45.570%. All related protocol's computation overhead is measured using the same method as shown in Table IV and Figure. 3.

C. Communication Overhead

The communication overhead shows how many bits were exchanged between the parties involved in the authentication procedure. As a result, we have evaluated the communication overhead of the suggested and related protocols [28], [29], [30], [31], [32]. Understanding that a protocol's registration phase for authentication is only executed once allows us to concentrate on the requests issued during the authentication phase when estimating the communication overhead. We compute it using 160-bit timestamps, passwords, XOR operations, arbitrary numbers, EC points, and identities. AES 128 has a 128-bit ciphertext/plaintext size. In comparison, a hash (SHA-256) will reserve 256 bits. In our proposed protocol the entities U_i, FN_j andCS exchanges four messages with each other. U_i sends (TID_{U_i}, R_1, α) to FN_j where $R_1 = r_1 P$, $TID_{U_i} = ID_{U_i} \oplus \bar{R}_1$ and $\alpha = h(ID_{U_i} \| TID_{U_i} \| \bar{R}_1 \| D_{ID_i})$. The number of bits exchanges between U_i and FN_j are (160+ 160+ 256) = 576 bits. Next FN_j exchange two messages $(TID_{U_i}, R_1, \alpha, TID_{FN_j}, R_3, \gamma)$ and (R_9, X_i) with CS and U_i respectively. Here the messages exchange between FN_j to CS are $R_1 = r_1 P$, $TID_{U_i} = ID_{U_i} \oplus \bar{R}_1$, $\alpha = h(ID_{U_i} \| TID_{U_i} \| \bar{R}_1 \| D_{ID_i})$, $R_3 = r_3 P$, $TID_{FN_j} = ID_{FN_j} \oplus \bar{R}_3$, $\gamma = h(ID_{FN_j} \| TID_{FN_j} \| \bar{R}_2 \| \bar{R}_3 \| D_{ID_j})$. Therefore, $(576 + 160 + 160 + 256) = 1152$ bits. The messages exchange between FN_j to U_i are (R_9, X_i). Here $R_9 = \hat{R}_3 \oplus (R_4 \| R_5)$, $X_i = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$ requires $(160 + 256) = 416$ bits to be transferred. The accumulative communication overhead of both FN_j to CS and FN_j to U_i is $(1152 + 416) = 1568$ bits. In the same way CS send message to FN_j are (R_7, R_8, R_9, X_j) where $R_7 = \hat{R}_1 \oplus R_4$, $R_8 = \hat{R}_2 \oplus (R_4 \| R_5)$, $R_9 = \hat{R}_3 \oplus (R_4 \| R_5)$, and $X_j = h(\hat{R}_1 \| \hat{R}_2 \| \hat{R}_3 \| SK_{i-k-j-cs})$. The amount of bits needed for the transmission are $(160 + 160 + 160 + 256) = 736$ bits. As a result, our protocol's overall communication overhead is $576 + 1152 + 416 + 736 = 2880$, and the communication overhead of our protocol is 45.454%, 34.782%, 19.642%, 26.229%, and 21.053% less than that of [28], [29], [30], [31], and [32] in that order. When compared to related protocols, our devised protocol reduces communication overhead by an average of 29.432%. All related protocol's communication overhead is measured using the same method as shown in Table V and Figure. 4.

TABLE V

Communication Overheads Comparison

$Protocol$	U_i	FN_j	CS	$Total\ Overhead$
Proposed	576 bits	1568 bits	736 bits	2880 bits
Tomar *et al.* [28]	1248 bits	2784 bits	1248 bits	5280 bits
Cui *et al.* [29]	992 bits	2080 bits	1344 bits	4416 bits
Jiang *et al.* [30]	736 bits	1920 bits	928 bits	3584 bits
Chen *et al.* [31]	576 bits	2400 bits	928 bits	3904 bits
Oh *et al.* [32]	736 bits	1824bits	1088 bits	3648 bits

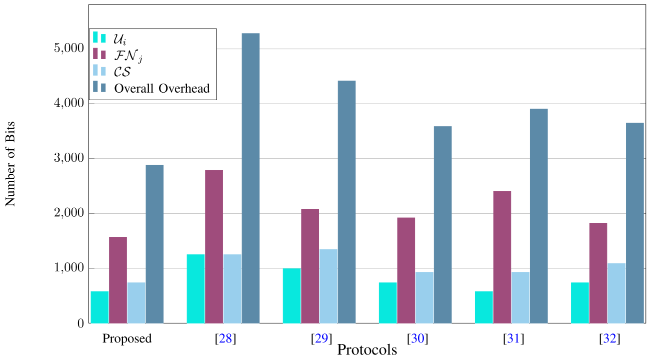

Fig. 4. Communication Overheads Analysis.

D. Security Comparison

Table VI Compares the security characteristics of the suggested and related protocols, using the references [28], [29], [30], [31], and [32]. This table demonstrates that the devised protocol assures the essential security characteristics, unlike the similar protocols [28], [29], [30], [31], [32] which do not achieve session key agreement and other security feathers.

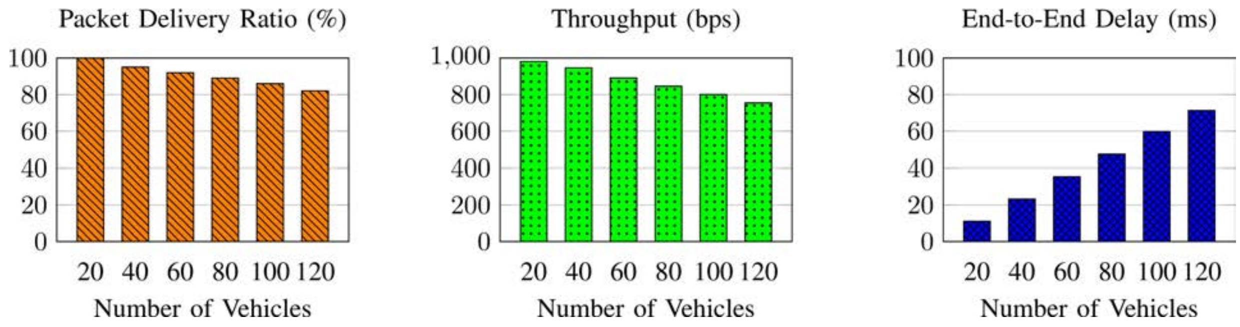

Fig. 5. Simulation results of Packet Delivery Ratio, Throughput, and End-to-End Delay.

TABLE VI

SECURITY ATTRIBUTES

Security Atrribute	*Proposed*	[28]	[29]	[30]	[31]	[32]
Mutual Authentication	Yes	Yes	Yes	Yes	Yes	Yes
User Impersonation Attack	Yes	Yes	Yes	No	Yes	No
Cloud Server Impersonation Attack	Yes	No	Yes	No	Yes	No
Session Key Agreement	Yes	Yes	Yes	Yes	Yes	No
User Anonymity	Yes	No	No	Yes	Yes	Yes
Untraceability	Yes	Yes	Yes	Yes	Yes	Yes
Perfect Forward Secrecy	Yes	Yes	No	Yes	Yes	No
Stolen Verifier Attack	Yes	No	No	No	No	No
Known Session Key Attack	Yes	Yes	No	Yes	No	No
Man-in-the-Middle Attack	Yes	Yes	Yes	No	Yes	Yes
Ephemeral leakage attack	Yes	No	Yes	No	No	No

Note:(Yes: Offers; No: Does not Offer)

TABLE VII

SIMULATION PARAMETERS

Parameters	*Value*
Number of vehicles	120
Operating system	Ubuntu 20.04
MAC protocol	MAC/802.11
Routing protocol	OLSR
Area of distribution	10 Km X 10 Km
Mobility model	Random walk 2d Mobility Model
Bandwidth	2Mbps
NS-3	Version 3.32
Traffic	Constant Bit rate
Radio Range	Random
Channel	Wireless
Simulation time	1800 seconds
Member speed	10 m/s

VI. SIMULATIONS USING NS-3

NS-3 is an open-source platform for simulating discrete-event networks. It offers several models to run different simulation tests, including TCP/UDP protocols, WiFi I modules, 6LoPAN, etc. NS 3 provides several libraries for margins and operations in various networks. Users can also use other programs, such as Python or C ++, to get simulation outcome.

A. Simulation Environment

Table VII provides the prerequisites for utilizing the NS3 simulator to simulate devised protocol and related protocols. Each vehicle should be 20 meters apart as it advances in a random route. 20 additional cars are added at each phase until 120 vehicles are present. All participating entities U_i, FN_j, and CS are considered in this topology. We also employ the values from [33] in accordance with the scheme and 802.11ah standard.In our protocol, the messages $M_1, M_3, M_4, and\ \ M_6$ have corresponding communication overhead of 576 bits, 1152 bits, 736 bits, and 416 bits. When choosing a car, the U_i user delivers a packet every time in 2 seconds. To assess the performance of our protocol, we use three performance metrics: end-to-end delay (ETE), throughput (T), and packet delivery ratio (PDR).

B. Packet Delivery Ratio

A key metric for assessing network communication efficiency is the packet delivery ratio (PDR), which is the ratio of packets successfully delivered to those transmitted. Figure.5 illustrates that PDR decreases as the number of cars increases, with the variation expanding to over 88 percent when tested with 120 vehicles. This decrease in PDR is mainly due to increased congestion from more vehicles in the simulation. Using the NS3 simulator, packets are transmitted over wireless media, and packets may be lost due to errors introduced by the wireless channel. Such losses occur when communication breaks down, influenced by the threshold set in the error model.

C. Throughput

Throughput, a key performance metric, measures the rate of successful data transfer in bits per second (bps) and is calculated using the formula $T = \frac{\sum(N_i \times Len_i)}{T_m}$, where N_i represents the number of packets, Len_i is the packet size, and T_m is the total time. Figure.5 illustrates that variations in throughput closely align with changes in the Packet Delivery Ratio (PDR). The primary driver of these fluctuations is the total packet count. More vehicles generally mean more packets are sent, which can reduce PDR due to increased noise and congestion, ultimately affecting throughput. Although improving PDR by reducing traffic may seem beneficial, this action can adversely impact throughput.

D. End-toEnd Delay

End-to-end delay (ETE) measures the average time taken for data to travel from source to destination. ETE is typically measured in milliseconds (ms) and is calculated with the formula $ETE = \frac{\sum_{i=1}^{n}(T_{ri} - T_{si})}{n}$, where n is the total number of received packets, T_{ri} is the packet's receiving time, and T_{si} is the transmission time of the i_{th} packet. As illustrated in Figure.5, ETE increases with the number of vehicles, exacerbated by greater distances and higher congestion.

VII. Conclusion

In collaboration with ITS, we developed a secure authentication and key agreement protocol for VANETs in smart cities, utilizing roadside units, fog nodes, and cloud servers to address significant security challenges. This protocol underwent rigorous security evaluations, proving to be effective against common threats such as impersonation attacks, while also enhancing intractability and ensuring vehicle anonymity. However, its implementation may face limitations related to scalability in densely populated networks and latency in real-time data processing, which could impact performance in dynamic environments. Future enhancements will focus on extending the protocol to support secure group communications across various network entities, aligning with technological advancements in the field. This adaptation is crucial for improving security and operational efficiency in emerging intelligent transportation systems.

Acknowledgment

The authors would like to thank the anonymous reviewers and associate editor for their valuable feedback on the article.

References

[1] L. Zhang, C. Hu, Q. Wu, J. Domingo-Ferrer, and B. Qin, "Privacy-preserving vehicular communication authentication with hierarchical aggregation and fast response," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2562–2574, Aug. 2016.

[2] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Netw. Comput. Appl.*, vol. 40, pp. 325–344, Apr. 2014.

[3] M. A. Saleem et al., "Provably secure conditional-privacy access control protocol for intelligent customers-centric communication in VANET," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1747–1756, Feb. 2024.

[4] H. Tahir, K. Mahmood, M. F. Ayub, M. A. Saleem, J. Ferzund, and N. Kumar, "Lightweight and secure multi-factor authentication scheme in VANETs," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 14978–14986, Nov. 2023.

[5] K. Mahmood et al., "Blockchain and PUF-based secure key establishment protocol for cross-domain digital twins in industrial Internet of Things architecture," *J. Adv. Res.*, vol. 62, pp. 155–163, Aug. 2024.

[6] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, "Distributed aggregate privacy-preserving authentication in VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 516–526, Mar. 2017.

[7] M. Rana, K. Mahmood, M. A. Saleem, F. Al-Turjman, M. S. Kolhar, and C. Altrjman, "Towards a provably secure authentication protocol for fog-driven IoT-based systems," *Appl. Sci.*, vol. 13, no. 3, p. 1424, Jan. 2023.

[8] M. A. Saleem, X. Li, K. Mahmood, T. Tariq, M. J. F. Alenazi, and A. K. Das, "Secure RFID-assisted authentication protocol for vehicular cloud computing environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 12528–12537, Sep. 2024.

[9] S. Roy, R. Bose, and D. Sarddar, "A fog-based DSS model for driving rule violation monitoring framework on the Internet of Things," *Int. J. Adv. Sci. Technol.*, vol. 82, pp. 23–32, Sep. 2015.

[10] X. Jia, D. He, N. Kumar, and K.-K.-R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Netw.*, vol. 25, no. 8, pp. 4737–4750, Nov. 2019.

[11] M. Ismail, S. Chatterjee, J. K. Sing, S. Kumari, and J. J. P. C. Rodrigues, "Designing anonymous key agreement scheme for secure vehicular ad-hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 11382–11396, Sep. 2024.

[12] L. Wei, J. Cui, H. Zhong, I. Bolodurina, and L. Liu, "A lightweight and conditional privacy-preserving authenticated key agreement scheme with multi-TA model for fog-based VANETs," *IEEE Trans. Depend. Sec. Comput.*, vol. 20, no. 1, pp. 422–436, Jan. 2023.

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

[14] Z. Li, Q. Miao, S. A. Chaudhry, and C.-M. Chen, "A provably secure and lightweight mutual authentication protocol in fog-enabled social Internet of Vehicles," *Int. J. Distrib. Sensor Netw.*, vol. 18, no. 6, Jun. 2022, Art. no. 155013292211043.

[15] I. Ul Haq, J. Wang, and Y. Zhu, "Secure two-factor lightweight authentication protocol using self-certified public key cryptography for multi-server 5G networks," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102660.

[16] P.-C. Zhao, Y.-H. Huang, D.-X. Zhang, L. Xing, H.-H. Wu, and J.-P. Gao, "CCP-federated deep learning based on user trust chain in social IoV," *Wireless Netw.*, vol. 29, no. 4, pp. 1555–1566, May 2023.

[17] A. Akbar, M. Ibrar, M. A. Jan, L. Wang, N. Shah, and H. H. Song, "SeAC: SDN-enabled adaptive clustering technique for social-aware Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 4827–4835, May 2023.

[18] J. Zheng, J. Shi, Q. He, E. Zhang, A. Hawbani, and L. Zhao, "An influence maximization-based hybrid advertising dissemination for Internet of Vehicles," *IEEE Netw. Lett.*, vol. 5, no. 4, pp. 218–222, Dec. 2023.

[19] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, "SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1810–1824, 2021.

[20] F. M. Salem, M. Safwat, R. Fathy, and S. Habashy, "AMAKAS: Anonymous mutual authentication and key agreement scheme for securing multi-server environments," *J. Cloud Comput.*, vol. 12, no. 1, p. 128, Aug. 2023.

[21] Y. Ma and Q. Cheng, "An anonymous and certificateless identity authentication protocol for mobile edge computing," *IEEE Syst. J.*, vol. 17, no. 4, pp. 5604–5615, Dec. 2023.

[22] C.-M. Chen, Z. Li, S. Kumari, G. Srivastava, K. Lakshmanna, and T. R. Gadekallu, "A provably secure key transfer protocol for the fog-enabled social Internet of Vehicles based on a confidential computing environment," *Veh. Commun.*, vol. 39, Feb. 2023, Art. no. 100567.

[23] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Inf. Syst.*, vol. 15, no. 9, pp. 1200–1215, Oct. 2021.

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

[25] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT)*, Amsterdam, The Netherlands. Berlin, Germany: Springer-Verlag, Apr. 2002, pp. 337–351.

[26] P. Rogaway and T. Shrimpton, "Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance," in *Proc. Int. Workshop Fast Softw. Encryption*, Delhi, India. Springer, Feb. 2004, pp. 371–388.

[27] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," Cryptol. ePrint Arch., Tech. Paper 2000/014, 2000. [Online]. Available: https://eprint.iacr.org/2000/014

[28] A. Tomar and S. Tripathi, "A Chebyshev polynomial-based authentication scheme using blockchain technology for fog-based vehicular network," *IEEE Trans. Mobile Comput.*, early access, Jan. 23, 2024, doi: 10.1109/TMC.2024.3357599.

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

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

[31] W.-C. Chen, Y.-T. Huang, and S.-D. Wang, "Provable secure group key establishment scheme for fog computing," *IEEE Access*, vol. 9, pp. 158682–158694, 2021.

[32] J. Oh, J. Lee, M. Kim, Y. Park, K. Park, and S. Noh, "A secure data sharing based on key aggregate searchable encryption in fog-enabled IoT environment," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 4468–4481, Nov. 2022.

[33] L. Tian, S. Deronne, S. Latré, and J. Famaey, "Implementation and validation of an IEEE 802.11ah module for ns-3," in *Proc. Workshop NS*, 2016, pp. 49–56.