*mathematics*

MDPI

*Article*

# A Secure Authentication Scheme with Local Differential Privacy in Edge Intelligence-Enabled VANET

**Deokkyu Kwon [1], Seunghwan Son [1], Kisung Park [2] and Youngho Park [1,*]**

[1] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea; kdk145@knu.ac.kr (D.K.); sonshawn@knu.ac.kr (S.S.)
[2] Department of Computer Engineering (Smart Security), Gachon University, Seongnam 13120, Republic of Korea; kisung@gachon.ac.kr
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

**Abstract:** Edge intelligence is a technology that integrates edge computing and artificial intelligence to achieve real-time and localized model generation. Thus, users can receive more precise and personalized services in vehicular ad hoc networks (VANETs) using edge intelligence. However, privacy and security challenges still exist, because sensitive data of the vehicle user is necessary for generating a high-accuracy AI model. In this paper, we propose an authentication scheme to preserve the privacy of user data in edge intelligence-enabled VANETs. The proposed scheme can establish a secure communication channel using fuzzy extractor, elliptic curve cryptography (ECC), and physical unclonable function (PUF) technology. The proposed data upload process can provide privacy of the data using local differential privacy and symmetric key encryption. We validate the security robustness of the proposed scheme using informal analysis, the Real-Or-Random (ROR) model, and the Scyther tool. Moreover, we evaluate the computation and communication efficiency of the proposed and related schemes using Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) software development kit (SDK). We simulate the practical deployment of the proposed scheme using network simulator 3 (NS-3). Our results show that the proposed scheme has a performance improvement of 10∼48% compared to the state-of-the-art research. Thus, we can demonstrate that the proposed scheme provides comprehensive and secure communication for data management in edge intelligence-enabled VANET environments.

**Keywords:** authentication; edge intelligence; local differential privacy; security analysis; vehicular ad hoc network

**MSC:** 68M12



**Citation:** Kwon, D.; Son, S.; Park, K.; Park, Y. A Secure Authentication Scheme with Local Differential Privacy in Edge Intelligence-Enabled VANET. *Mathematics* **2024**, *12*, 2383. https://doi.org/10.3390/math12152383

Academic Editors: Cheng-Chi Lee and Dinh-Thuan Do

Received: 11 July 2024
Revised: 30 July 2024
Accepted: 30 July 2024
Published: 31 July 2024



## 1. Introduction

Edge intelligence is a convergence technology of edge computing and artificial intelligence (AI) [1,2]. AI technology requires a large volume of user data to generate learning models. In edge computing networks, service providers can collect the real-time information from the network edge. Therefore, the combination of edge computing technology and AI can amplify the synergy through the real-time data collection and reflection of regional characteristics in AI models. Through these advantages, researchers have applied edge intelligence into vehicle services, such as vehicular ad hoc networks (VANETs) [3,4]. In edge intelligence-enabled VANET environments, users can receive improved vehicular services using AI, such as localized autonomous driving, accident prediction, and personalized entertainment experiences. To provide these services, a large volume of vehicle users' personal data is necessary for training AI models [5]. If an adversary obtains these data (e.g., driving habits and history, call and messaging history in infotainment systems), it can cause serious security problems. Although the data are securely encrypted using various cryptography methods, the central server can still access user data. This can cause user

privacy, anonymity and traceability problems. Therefore, it is important to strengthen the de-identification of user data in edge intelligence-enabled VANET environments, while preserving the confidentiality and availability of the data.

Differential privacy [6] is a de-identification technology that can provide privacy by adding noise or shuffling the data. The main advantage of differential privacy is that it simultaneously satisfies privacy protection and information analysis. This is possible because the statistical properties of the information can be maintained even when differential privacy is applied. However, these characteristics can still present security vulnerabilities to attackers. For example, if an adversary obtains the differential privacy-based data due to a low security level, it can generate a similar AI model. This can threaten the edge intelligence-enabled VANET network, because the adversary can infer the behavior patterns of vehicle users. This can compromise the anonymity and untraceability of vehicle users. Such vulnerabilities highlight the need for a robust authentication scheme to protect the differential privacy-based data.

In this paper, we propose a secure authentication scheme designed to preserve user data and ensure the privacy of generated data in edge intelligence-enabled VANET environments. The proposed scheme supports mutual authentication between edge servers and vehicles in VANET environments, as well as differential privacy-based data uploads. The proposed scheme provides a secure and efficient key agreement using fuzzy extractors [7], biometric information, and elliptic curve cryptography (ECC) [8]. Moreover, the proposed scheme can prevent potential security attacks, such as machine learning attacks, by utilizing physically unclonable function (PUF). Thus, the proposed mutual authentication process provides a high level of security to prevent adversaries from accessing user data. In the data upload process, users can achieve data privacy and anonymity using symmetric key encryption and differential privacy. By integrating these technologies, the proposed scheme can ensure not only data integrity and confidentiality during message transmission, but also user privacy from unauthorized access. The key contributions of our proposed scheme are as follows:

- We propose a secure authentication scheme for edge intelligence-enabled VANET environments. The proposed scheme can provide a secure communication between edge nodes and vehicles using fuzzy extractors, biometric information, and ECC. To ensure the robust security for edge nodes, the proposed scheme utilizes PUF technology when generating the secret keys.
- We provide a secure data upload process using the session key and local differential privacy technology [6]. Thus, the proposed scheme can ensure secure message transmission and data collection through the encryption of de-identification data. This approach can provide secure and efficient data management for edge intelligence-enabled VANET environments.
- We perform various analyses to prove the security robustness of the proposed scheme, such as informal analysis, as well as using the "Real-Or-Random (ROR) model [9]", and the Scyther tool [10,11]. Moreover, we conduct a simulation study using "Network Simulator (NS)-3 [12]".
- We compare the computation and communication overheads of the proposed scheme with the other related schemes using "Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) software development kit (SDK) [13]".

The rest of our paper is structured as follows. In "Related Works" (Section 2), various research is introduced for edge intelligence-based VANETs. In "Preliminaries" (Section 3), the system model, threat model, and various security technologies are introduced. In "Proposed Scheme" (Section 4), the detailed scheme is introduced. In "Security Analysis" (Section 5), an informal analysis is conducted, and the ROR model and Scyther tool are used to prove the security robustness of the proposed scheme. In "Performance Analysis" (Section 6), the comparison and NS-3 simulation studies are performed to verify the practical deployment of the proposed scheme. In "Conclusions" (Section 7), we conclude and summarize our paper.

## 2. Related Works

Edge intelligence is an emerging topic, aiming to expand advanced services using edge computing technology [14–21]. In 2019, Zhou et al. [14] proposed the basic concept of edge intelligence. They argued that it is important to perform computation tasks on edge nodes to solve centralization, a major challenge in cloud computing. Moreover, they demonstrated that edge intelligence has advantages over cloud intelligence because real-time information is generated at the edge of the network system. Zhou et al. also introduced three types of distributed training architecture: centralized, decentralized, and hybrid methods. Deng et al. [15] discussed edge intelligence in terms of utilizing a large amount of data generated from the edge. Thus, they distinguished the usage of edge intelligence into "AI for edge (AFE)" and "AI on edge (AOE)". In their paper, AFE utilizes AI to identify and improve challenges with edge computing devices in the network system. Therefore, AFE is an assist concept for optimized edge computing. On the other hand, AOE is a concept defined to maximize edge computing services. Therefore, the edge node collects information on the network, creates a learning model based on it, and uses it to improve services in AOE. In 2021, Qi et al. [16] proposed a resource management architecture to achieve a sufficient edge intelligent service for future vehicular networks. The network architecture is composed of data perception, machine learning, edge access, intelligent control, and application layers. Thus, various sensors in the data perception layer collect surrounding data and send them to machine learning layer. To conduct efficient machine learning, cloud computing is utilized for reducing the load in edge nodes. In Qi et al.'s architecture, base stations and RSUs are in the edge access layer to provide a wide range of edge resources. In the application layer, vehicles provide various network services, including autonomous driving, smart parking, and traffic notifications using edge intelligence technology. In 2023, Gong et al. [17] introduced the integration of edge intelligence and an intelligent transportation system (ITS). They argued that edge intelligence technology can maintain low latency and energy efficiency, and reduce the load on the backbone network. From that, Gong et al. described the basic structure of edge intelligence-enabled ITS. Moreover, they discussed the security issues in edge intelligence-enabled ITSs, such as data leakage, the preservation of privacy, and the sensitivity of vehicle data. Thus, they emphasized the necessity of a privacy policy (e.g., General Data Protection Regulation (GDPR)), differential privacy, and various encryption methods. In addition, various other papers have proposed and highlighted the importance of edge intelligence [18–21]. They also emphasized the necessity of security protocols and data anonymization to provide convenient edge services. These contributions collectively underline the critical role of edge intelligence in enhancing computational efficiency and service quality at the network's edge, while also addressing essential concerns related to data security and privacy.

In VANET environments where traffic data are distributed on a large scale, edge intelligence must consider security in terms of wireless communication, distributed computing, and data management. Therefore, research on mutual authentication in edge computing environments is necessary. In 2019, Jia et al. [22] proposed a mutual authentication protocol for mobile edge computing (MEC) environments. They argued that MEC environments can suffer from security problems because MEC devices are deployed by various service providers. Thus, Jia et al. proposed an authentication scheme using ECC and bilinear pairings. Bagga et al. [23] suggested an authentication protocol for Internet of vehicle (IoV) environments. In their protocol, anonymity and untraceability are achieved using a pseudo identity-based authentication method. Ke et al. [24] proposed an authentication scheme for smart healthcare systems. In their system model, software defined networking (SDN) technology is utilized to monitor data flow such as the authentication requirements of users. They also used bilinear pairings to achieve a high level of security for the authentication protocol. In 2023, Seifelnasr et al. [25] proposed a privacy-preserving authentication protocol using the computation capability of edge nodes for IoT environments. They utilized zero knowledge proof technology and elliptic curve Diffie–Hellman (ECDH) to

ensure the anonymity of IoT devices and the robust establishment of a session key. In 2024, Yadav et al. [26] proposed an authentication protocol for efficient and secure communication between vehicle and infrastructure. Kumar and Om [27] proposed an authentication protocol for fog-enabled VANET environments. In their protocol, vehicle users access the network through a third-party authentication process that leads to vehicle–fog–TA. While these schemes [22–26] provide various convenient services to users, they suffer from high computational overheads due to the use of bilinear pairings and elliptic curve-based signatures. Furthermore, these edge computing-based security schemes [22–26] do not adequately address the privacy of user data, which is crucial in edge intelligence-enabled VANET environments. Here, the server and edge nodes require large amounts of user data, including sensitive information. Thus, the proposed scheme aims to establish a secure communication channel between vehicles and edge nodes while integrating local differential privacy and user authentication to ensure robust user privacy. The summary of related works [22–27] are shown in Table 1.

**Table 1.** Summary of the proposed scheme and related schemes.

| Year | Scheme | Contributions | Limitations |
|---|---|---|---|
| 2019 | [22] | • Proposed a network model for communicating MEC devices and users<br>• Proposed an identity-based mutual authentication protocol for MEC environments<br>• Using bilinear pairings and ECC | • Cannot prevent impersonation and ESL attacks<br>• Does not ensure perfect forward secrecy<br>• Large computation costs: bilinear pairings |
| 2021 | [23] | • Proposed an authentication and key agreement scheme for ITS environments<br>• Using ECC and hash functions | • Cannot prevent impersonation and physical attacks |
| 2022 | [24] | • Proposed a system model for medical environments using SDN technology<br>• Proposed an authentication scheme between fog nodes and IoT devices<br>• Using bilinear pairings and ECC | • Large computation costs: bilinear pairings |
| 2023 | [25] | • Proposed an authentication scheme considering the computation capacity of edge and IoT devices<br>• Introduced IoT, edge, and cloud layers-based network architecture<br>• Using ECC and ECC-based signature | • Cannot ensure untraceability<br>• Requires high computation costs using ECC-based signature |
| 2024 | [26] | • Introduced a fog-based network model in VANET environments<br>• Proposed an authentication and key agreement scheme using message broadcast methods.<br>• Utilized ECC and hash functions | • Requires a high communication costs using message broadcast method to authenticate vehicle. |
| 2024 | [27] | • Proposed a system model that a fog node manages regional RSUs.<br>• Proposed an authentication scheme to establish session key between vehicle and fog node.<br>• Utilized ECC, symmetric key encryption and hash functions | • Central server must be involved in authentication process between fog nodes and vehicles<br>• Requries high communication overhead due to third-party authentication |
| - | Proposed | • Proposes an authentication and key agreement scheme between the edge node and vehicle using ECC, PUF, and hash functions<br>• Provides a secure login process using biometrics and fuzzy extractors<br>• Proposes a data collection phase using local differential privacy and symmetric key encryption to achieve privacy of user data | |

## 3. Preliminaries

### *3.1. System Model*

The proposed system model for edge intelligence-enabled VANET environments consists of trusted authority (TA), a cloud server, an edge node, and a vehicle. Figure 1 shows the proposed system model and the details are as follows:

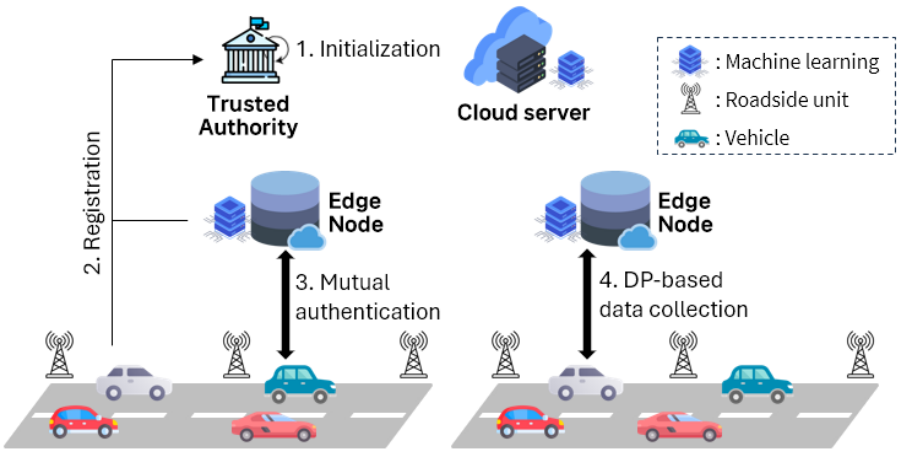


**Figure 1.** System model.

#### 3.1.1. Trusted Authority

TA manages the proposed network system by initializing and publishing public information such as ECC, hash function, and global public key. Moreover, TA performs the registration process and stores the sensitive data of the cloud server, edge node, and vehicle. TA has a large amount of computation and storage resources.

#### 3.1.2. Cloud Server

The cloud server controls the entire VANET service and data based on enormous computing and storage resources. Additionally, the cloud server creates a large AI model using vehicle data sent by edge nodes.

#### 3.1.3. Edge Node

An edge node is a infrastructure controlled by TA, which manages services and collects information for vehicles in a specific area through RSUs. Additionally, the edge node collects and learns local information sent by vehicles based on sufficient computing and storage resources to create a local AI model. Edge nodes can use this edge intelligence to provide improved VANET services to vehicles. Moreover, edge nodes help to create a global AI model for the overall VANET service by uploading some information about the vehicle to the cloud server.

#### 3.1.4. Vehicle

Vehicles can receive various VANET services such as AI-based route guidance, entertainment, and accident prediction through mutual authentication with edge nodes. Additionally, vehicles upload some of their driving data to continuously improve VANET services and enhance the accuracy of accident prediction. Because the uploaded data are safely masked using local differential privacy, edge nodes cannot identify the exact information of individual vehicles. In the proposed scheme, the vehicle has limited computational and storage resources.

### *3.2. Threat Model*

In the proposed scheme, we utilize "Dolev-Yao (DY) [28]" and "Canetti-Krawczyk (CK) [29]" network models. In the DY model, the adversary has access to messages on public channels. Therefore, the adversary can eavesdrop on, insert, capture, and delete messages transmitted via public channels. In the CK network model, the adversary can access secret credentials. Thus, the adversary can obtain a revealed master key of the TA and ephemeral secret value in the proposed scheme. Using the DY and CK network models, the adversary can process the following security attacks:

- The adversary can reveal the verification table and try to compute sensitive parameters [30].
- The adversary can obtain secret parameters and try to disguise itself as a legitimate vehicle [31].
- The adversary can be a privileged insider to compute identity and password of vehicle.
- The adversary can perform various security attacks such as man-in-the-middle, ephemeral secret leakage, replay, and insider attacks.

### *3.3. Elliptic Curve Cryptography*

Elliptic curve cryptography (ECC) [8] is a cryptosystem that implements cryptographic characteristics using elliptic curves. To utilize ECC in a security system, we must select a large finite field $\mathbb{F}_p$, large prime integer $p$, $q$, and ECC parameter $w$, $v$. Then, we can generate an elliptic curve $E(w, v) : y^2 = x^3 + wx + v$ $(4w^3 + 27v^2 \neq 0)$. Since the point on the elliptic curve satisfies the addition group, we specify a base point $P$. Therefore, ECC satisfies the following equation for an integer $n \in \mathbb{Z}_q$. Moreover, we introduce the mathematical security of ECC as follows:

$$n \cdot P = P + P + P + \ldots + P \ (n \text{ times})$$

- Elliptic curve discrete logarithm (ECDL) problem: A mathematical problem to compute $n \in \mathbb{Z}_q$ when $n \cdot P$ is given.
- Elliptic curve decisional Diffie–Hellman (ECDDH) problem: A mathematical problem to grant the equality of $n \cdot s \cdot P$ and $t \cdot P$ when $n$, $s$, and $t$ is allowed.
- Elliptic curve computational Diffie–Hellman (ECCDH) problem: A mathematical problem to compute $n \cdot s \cdot P$ when $n \cdot P$ and $s \cdot P$ are allowed.

### *3.4. Physically Unclonable Function*

Physical unclonable function (PUF) is a technology that implements a one-way function in hardware. PUF is performed as $Res = PUF(Cha)$, where $Cha$ is a input value "Challenge" and $Res$ is an output value "Response". We introduce the properties of an ideal PUF as follows:

- PUF is a hardware circuit, which cannot replicate or interpret the detailed structure.
- Since PUF is implemented uniquely in each hardware, different outputs are produced even if the same input is input.
- The output value of PUF cannot be predicted.
- PUF is easy to implement and estimate.

### *3.5. Fuzzy Extractor*

Fuzzy extractor [7] is a method to utilize biometric information of users as a security parameter. Unlike identity and password, biometrics, e.g., fingerprint and iris information, are detected by a sensor. Thus, the input data can change slightly depending on the surrounding environments. Nevertheless, this information must be constant to be used as the security parameters [32]. Fuzzy extractor can correct this noise-based information to original data using the helper string. Fuzzy extractor is composed of two algorithms, i.e., "generation ($Gen(.)$)" and "reproduce ($Rep(.)$)".

- $Gen(Bio_{VE-i}) = (eb_{VE-i}, hs_{VE-i})$: After executing the probability algorithm $Gen(.)$, we can obtain a string $eb_{VE-i}$ and helper string $hs_{VE-i}$. We utilize $eb_{VE-i}$ as a secret parameter for the proposed scheme.
- $Rep(Bio'_{VE-i}, hs_{VE-i}) = (eb_{VE-i})$: After conducting the deterministic algorithm $Rep(.)$ with helper string $hs_{VE-i}$, we can obtain the secret parameter $eb_{VE-i}$.

### *3.6. Local Differential Privacy*

Differential privacy is a technique that preserves the privacy of individual users while maintaining the statistical trends of the overall user dataset. This technique can be implemented by introducing randomness to individual responses through mechanisms like

randomized response, or by adding various types of noise such as Laplace, Gaussian, or exponential noise to the original data. In 2006, Dwork et al. [6] proposed epsilon-differential privacy to quantify the level of privacy preservation provided by different differential privacy techniques. Definition 1 illustrates $\epsilon$-differential privacy, and Definition 2 illustrates differential privacy using the Laplace Probability Density Function (PDF).

**Definition 1.** *$\epsilon$-differential privacy: For a randomized algorithm $\mathbb{A}$, it is differentially private ($\epsilon$) when two datasets $D$ and $D'$ have a difference in one element. $S$ is subset of output using $\mathbb{A}$.*

$$Pr[\mathbb{A}(D) \in S] \leq e^{\epsilon} \cdot Pr[\mathbb{A}(D') \in S]$$

**Definition 2.** *To achieve $\epsilon$-differential privacy, the Laplace mechanism adds noise drawn from the Laplace distribution to the output of a function $f$. The Laplace distribution with scale parameter $\lambda = \frac{\Delta f}{\epsilon}$ has the following probability density function:*

$$Lap(x|\lambda) = \frac{1}{2\lambda} e^{-\frac{|x|}{\lambda}}$$

Differential privacy applies privacy protection at the central server level, which still leaves a possibility for personal data leakage. Local differential privacy, on the other hand, ensures privacy by adding noise to the data on the user's end device before sending them to the server, thereby achieving better privacy protection. Figure 2 illustrates the difference between general differential privacy and local differential privacy.

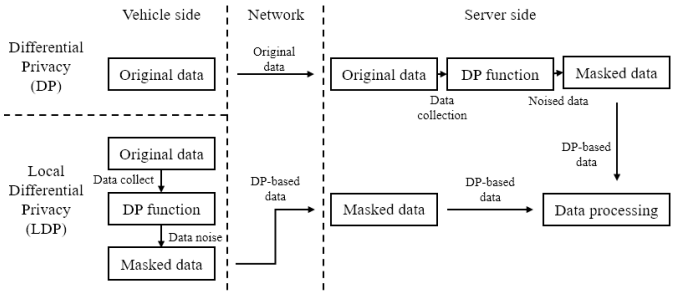


**Figure 2.** General differential privacy and local differential privacy.

## 4. Proposed Scheme

In this section, we propose a mutual authentication and data collection scheme for edge intelligence-enabled VANET environments. The proposed scheme consists of initialization, registration, login and authentication, and differential privacy-based data collection phases. Notations and descriptions in the proposed scheme are shown in Table 2. Figure 3 indicates the flowchart of the proposed scheme, and the details are as follows:

**Table 2.** Notations and descriptions.

| Notation | Explanation |
|---|---|
| $ID_{VE-i}, ID_{ED-k}$, | Real identity of vehicle and edge node |
| $PW_{VE-i}$ | Password of vehicle |
| $Bio_{VE-i}$ | Biometric information of vehicle |
| $PID_{VE-i}$ | Pseudo identity of vehicle |
| $hs_{VE-i}$ | Helper string |
| $rs_m$ | Random nonce |
| $ts_m$ | Timestamp |
| $Pub_m$ | Public key of an entity $m$ |
| $sk_m$ | Secret key of an entity $m$ |
| $Gen(.)$ | Generation algorithm of fuzzy extractor |
| $Rep(.)$ | Reproduce algorithm of fuzzy extractor |
| $PUF(.)$ | PUF operator |
| $SK$ | Session key |
| $\cdot$ | Multiplication operator |
| $\oplus$ | XOR operator |
| $h(.)$ | Hash function |
| $\|\|$ | Concatenation operator |

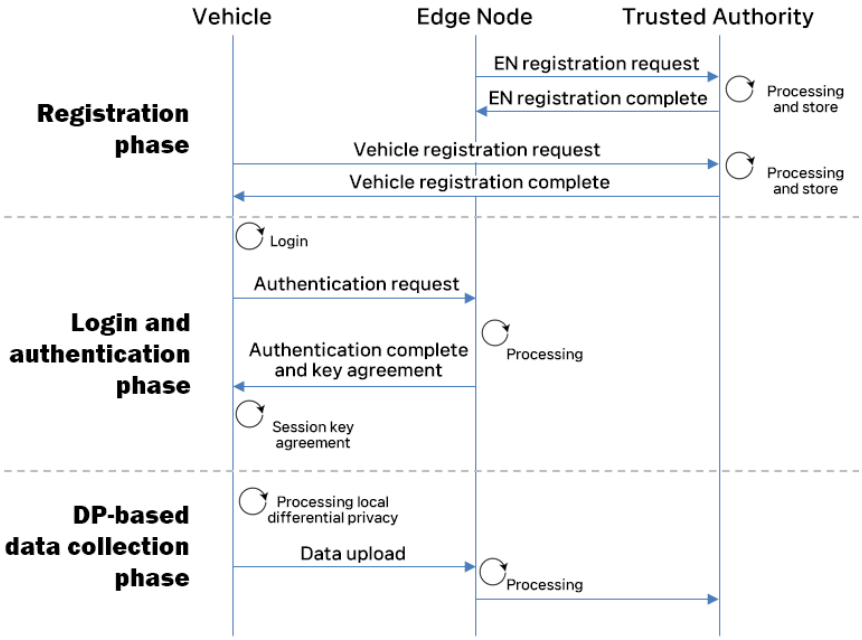


**Figure 3.** Flowchart of the proposed scheme.

### *4.1. Initialization Phase*

In this phase, TA performs an initial setup to organize the proposed network. TA selects large prime number $p, q$ and picks $w, v \in F_p$. Then, TA generates an elliptic curve $E(w,v) : y^2 = x^3 + wx + v \; mod \; q$. Furthermore, TA selects a generator $P$ from $E(w,v)$ and picks a master key $mk_{TA}$ to compute the public key $Pub_{TA} = mk_{TA} \cdot P$. TA selects a hash function $h(.)$ and publishes $\{E(w,v), P, h(.), Pub_{TA}, p, q\}$.

### *4.2. Registration Phase*

To participate in the proposed network environments, edge nodes and vehicles must process the registration phase by sending their information to TA. After registering these entities, TA returns a secret credential using a secure channel. The details are as follows:

#### 4.2.1. Edge Node Registration

**RE1:** To register in the proposed network system, the edge node $E_k$ selects its own identity $ID_{ED-k}$ and picks a random number $ns_{ED-k}$. Then, $E_k$ computes $h(ID_{ED-k} \parallel ns_{ED-k})$ and sends $\{ID_{ED-k}, h(ID_{ED-k} \parallel ns_{ED-k})\}$ to the TA via a secure channel.

**RE2:** TA first checks the validity of $ID_{ED-k}$ and generates $ns_{TA-Ek}$. Then, TA computes $h(h(ID_{ED-k} \parallel ns_{ED-k}) \parallel ns_{TA-Ek} \parallel mk_{TA})$ and stores $\{ID_{ED-k}, h(ID_{ED-k} \parallel ns_{ED-k})\}$ in its secure database. TA returns $\{h(h(ID_{ED-k} \parallel ns_{ED-k}) \parallel ns_{TA-ED} \parallel mk_{TA})\}$ to $E_k$ through a secure channel.

**RE3:** $E_k$ computes $psk_{ED-k} = h(h(ID_{ED-k} \parallel ns_{ED-k}) \parallel ns_{TA-ED} \parallel mk_{TA})$, $PUF(psk_{ED-k}) = usk_{ED-k}$ using PUF function, $Gen(usk_{ED-k}) = (eusk_{ED-k}, hs_{ED-k})$ using fuzzy extractor, and $sk_{ED-k} = h(eusk_{ED-k} \parallel ID_{ED-k})$. $E_k$ keeps $sk_{ED-k}$ as a secret key and computes public key $Pub_{ED-k} = sk_{ED-k} \cdot P$. $E_k$ stores $\{psk_{ED-k}, Pub_{ED-k}, hs_{ED-k}\}$ in its database.

#### 4.2.2. Vehicle Registration

**RV1:** The user of a vehicle $VE_i$ selects their own identity $ID_{VE-i}$, password $PW_{VE-i}$, and biometrics $Bio_{VE-i}$. Then, $VE_i$ picks a random number $ns_{VE-i}$ and computes $Gen(Bio_{VE-i}) = (eb_{VE-i}, hs_{VE-i})$ using fuzzy extractor, $MID_{VE-i} = h(ns_{VE-i} \parallel ID_{VE-i} \parallel eb_{VE-i})$. $VE_i$ sends a registration request message $\{ID_{VE-i}, MID_{VE-i}, ns_{VE-i}\}$ to the TA via a secure channel.

**RV2:** TA checks the validity of $ID_{VE-i}$ and generates $ns_{TA-Vi}$ to compute $CMK_{TA-Vi} = h(MID_{VE-i} \parallel ns_{TA-Vi} \parallel mk_{TA})$, $PID_{VE-i} = h(ID_{VE-i} \parallel mk_{TA})$, and $SID_{VE-i} = ID_{VE-i} \oplus h(mk_{TA} \parallel PID_{VE-i} \parallel ns_{TA-Vi})$. TA stores $\{PID_{VE-i}, SID_{VE-i}, ns_{TA-Vi}\}$ and sends a return message $\{PID_{VE-i}, CMK_{TA-Vi}\}$ to $VE_i$ through a secure channel.

**RV3:** $VE_i$ computes its secret key $sk_{VE-i} = h(CMK_{TA-Vi} \parallel eb_{VE-i})$ and public key $Pub_{VE-i} = sk_{VE-i} \cdot P$. Then, $VE_i$ computes $zns_{VE-i} = ns_{VE-i} \oplus h(ID_{VE-i} \parallel eb_{VE-i})$, $zPID_{VE-i} = PID_{VE-i} \oplus h(ns_{VE-i} \parallel eb_{VE-i} \parallel PW_{VE-i})$, $zCMK_{TA-Vi} = CMK_{TA-Vi} \oplus h(MID_{VE-i} \parallel PID_{VE-i} \parallel ID_{VE-i})$, $zPub_{VE-i} = Pub_{VE-i} \oplus h(sk_{VE-i} \parallel eb_{VE-i} \parallel ns_{VE-i})$, and $V_{VE-i} = h(sk_{VE-i} \parallel Pub_{VE-i} \parallel ns_{VE-i} \parallel PID_{VE-i} \parallel MID_{VE-i})$. $VE_i$ stores $\{zns_{VE-i}, zPID_{VE-i}, zCMK_{TA-Vi}, zPub_{VE-i}, V_{VE-i}, hs_{VE-i}\}$ in its memory.

### *4.3. Login and Authentication Phase*

To receive edge intelligence services, a registered vehicle submits its information to complete the login process. Subsequently, the vehicle selects a fresh value, encrypts the information using a public key, and attempts to establish a session key agreement with the edge node. The edge node uses the public key and PUF technologies for mutual authentication, ensuring high security. Algorithms 1 and 2, and Figure 4 present the proposed login and authentication phase. The detailed process is as follows:

**Algorithm 1:** Login and authentication: Vehicle

**Input:** $ID'_{VE-i}$, $PW'_{VE-i}$, $Bio'_{VE-i}$
**Output:** $SK_{EK-Vi}$, $\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}$

Compute $Rep(Bio'_{VE-i}, hs_{VE-i}) = eb'_{VE-i}$
$ns'_{VE-i} = zns_{VE-i} \oplus h(ID'_{VE-i} \parallel eb'_{VE-i})$
$PID'_{VE-i} = zPID_{VE-i} \oplus h(ns'_{VE-i} \parallel eb'_{VE-i} \parallel PW'_{VE-i})$
$MID'_{VE-i} = h(ns'_{VE-i} \parallel ID'_{VE-i} \parallel eb'_{VE-i})$
$CMK'_{TA-Vi} = zCMK_{TA-Vi} \oplus h(MID'_{VE-i} \parallel PID'_{VE-i} \parallel ID'_{VE-i})$
$sk'_{VE-i} = h(CMK'_{TA-Vi} \parallel eb'_{VE-i})$
$Pub'_{VE-i} = zPub_{VE-i} \oplus h(sk'_{VE-i} \parallel eb'_{VE-i} \parallel ns'_{VE-i})$
$V'_{VE-i} = h(sk'_{VE-i} \parallel Pub'_{VE-i} \parallel ns'_{VE-i} \parallel PID'_{VE-i} \parallel MID'_{VE-i})$
**if** $(V'_{VE-i} \stackrel{?}{=} V_{VE-i})$
**then**
  Pick a random nonce $rs_i$ and timestamp $ts_i$
  Compute $TP_{mv-i} = rs_i \cdot P$
  $TK_{mv-i} = rs_i \cdot Pub_{ED-k}$
  $VS_{mv-a} = PID_{VE-i} \oplus h(ts_i \parallel TK_{mv-i})$
  $VS_{mv-b} = h(Pub_{VE-i} \parallel TK_{mv-i} \parallel PID_{VE-i} \parallel ts_i)$
  Send $\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}$
  Wait for return message
  **if** $(|ts_k - ts_c| < \Delta t)$
  **then**
    Computes $TK'_{me-k} = sk_{VE-i} \cdot TP_{me-k}$
    $SK_{Ek-Vi} = h(TK'_{me-k} \parallel ts_i \parallel ts_k \parallel PID_{VE-i} \parallel Pub_{VE-i})$
    $ES'_{me-a} = h(SK_{Ek-Vi} \parallel TK_{me-k} \parallel PID_{VE-i} \parallel Pub_{VE-i})$
    **if** $(ES_{me-a} \stackrel{?}{=} ES'_{me-a})$
    **then**
      Authentication and key agreement success;
      Established : Session key $SK_{EK-Vi}$
    **else**
      Authentication failure;
    **end**
  **else**
    Authentication failure;
  **end**
**else**
  Login failure;
**end**

**Algorithm 2:** Login and authentication: Edge node

**Input:** $\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}$
**Output:** $SK_{Vi-Ek}$, $\{TP_{me-k}, ES_{me-a}, ts_k\}$

Receive $\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}$ **if** $(|ts_i - ts_c| < \Delta t)$ **then**
  Compute $PUF(psk_{ED-k}) = usk_{ED-k}$
  $Rep(usk_{ED-k}, hs_{ED-k}) = eusk_{ED-k}$
  $sk_{ED-k} = h(eusk_{ED-k} \| ID_{ED-k})$
  $TK'_{mv-i} = TP_{mv-i} \cdot sk_{ED-k}$
  $PID'_{VE-i} = VS_{mv-a} \oplus h(ts_i \| TK'_{mv-i})$
  Retrieve $Pub_{VE-i}$
  **if** $VS_{mv-b} \stackrel{?}{=} h(Pub'_{VE-i} \| TK'_{mv-i} \| PID'_{VE-i} \| ts_i)$
  **then**
    Generate $rs_k$, $ts_k$
    Compute $TP_{me-k} = rs_k \cdot P$
    $TK_{me-k} = rs_k \cdot Pub_{VE-i}$
    $SK_{Vi-Ek} = h(TK_{me-k} \| ts_i \| ts_k \| PID_{VE-i} \| Pub_{VE-i})$
    $ES_{me-a} = h(SK_{Vi-Ek} \| TK_{me-k} \| PID_{VE-i} \| Pub_{VE-i})$
    Send $\{TP_{me-k}, ES_{me-a}, ts_k\}$
  **else**
    Authentication failure;
  **end**
**else**
  Authentication failure;
**end**

| Vehicle ($VE_i$) | Edge node ($E_k$) |
|---|---|
| Inputs $ID'_{VE-i}$, $PW'_{VE-i}$, and biometrics $Bio'_{VE-i}$ | |
| Computes $Rep(Bio'_{VE-i}, hs_{VE-i}) = eb'_{VE-i}$ | |
| $ns'_{VE-i} = zns_{VE-i} \oplus h(ID'_{VE-i} \| eb'_{VE-i})$ | |
| $PID'_{VE-i} = zPID_{VE-i} \oplus h(ns'_{VE-i} \| eb'_{VE-i} \| PW'_{VE-i})$ | |
| $MID'_{VE-i} = h(ns'_{VE-i} \| ID'_{VE-i} \| eb'_{VE-i})$ | |
| $CMK'_{TA-Vi} = zCMK_{TA-Vi} \oplus h(MID'_{VE-i} \| PID'_{VE-i} \| ID'_{VE-i})$ | |
| $sk'_{VE-i} = h(CMK'_{TA-Vi} \| eb'_{VE-i})$ | |
| $Pub'_{VE-i} = zPub_{VE-i} \oplus h(sk'_{VE-i} \| eb'_{VE-i} \| ns'_{VE-i})$ | |
| $V'_{VE-i} = h(sk'_{VE-i} \| Pub'_{VE-i} \| ns'_{VE-i} \| PID'_{VE-i} \| MID'_{VE-i})$ | |
| Checks $V'_{VE-i} \stackrel{?}{=} V_{VE-i}$ | |
| Picks a random nonce $rs_i$ and timestamp $ts_i$ | |
| Computes $TP_{mv-i} = rs_i \cdot P$ | |
| $TK_{mv-i} = rs_i \cdot Pub_{ED-k}$ | |
| $VS_{mv-a} = PID_{VE-i} \oplus h(ts_i \| TK_{mv-i})$ | |
| $VS_{mv-b} = h(Pub_{VE-i} \| TK_{mv-i} \| PID_{VE-i} \| ts_i)$ | |
| $\xrightarrow{\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}}$ | |
| | Checks $\|ts_i - ts_c\| < \Delta t$ |
| | Computes $PUF(psk_{ED-k}) = usk_{ED-k}$ |
| | $Rep(usk_{ED-k}, hs_{ED-k}) = eusk_{ED-k}$ |
| | $sk_{ED-k} = h(eusk_{ED-k} \| ID_{ED-k})$ |
| | $TK'_{mv-i} = TP_{mv-i} \cdot sk_{ED-k}$ |
| | $PID'_{VE-i} = VS_{mv-a} \oplus h(ts_i \| TK'_{mv-i})$ |
| | Retrieves $Pub_{VE-i}$ |
| | Checks $VS_{mv-b} \stackrel{?}{=} h(Pub'_{VE-i} \| TK'_{mv-i} \| PID'_{VE-i} \| ts_i)$ |
| | Generates $rs_k$ and $ts_k$ |
| | Computes $TP_{me-k} = rs_k \cdot P$ |
| | $TK_{me-k} = rs_k \cdot Pub_{VE-i}$ |
| | $SK_{Vi-Ek} = h(TK_{me-k} \| ts_i \| ts_k \| PID_{VE-i} \| Pub_{VE-i})$ |
| | $ES_{me-a} = h(SK_{Vi-Ek} \| TK_{me-k} \| PID_{VE-i} \| Pub_{VE-i})$ |
| | $\xleftarrow{\{TP_{me-k}, ES_{me-a}, ts_k\}}$ |
| Checks $\|ts_k - ts_c\| < \Delta t$ | |
| Computes $TK'_{me-k} = sk_{VE-i} \cdot TP_{me-k}$ | |
| $SK_{Ek-Vi} = h(TK'_{me-k} \| ts_i \| ts_k \| PID_{VE-i} \| Pub_{VE-i})$ | |
| $ES'_{me-a} = h(SK_{Ek-Vi} \| TK_{me-k} \| PID_{VE-i} \| Pub_{VE-i})$ | |
| Checks $ES_{me-a} \stackrel{?}{=} ES'_{me-a}$ | |

**Figure 4.** Login and authentication phase of the proposed scheme.

**LA1:** $VE_i$ inputs $ID'_{VE-i}$, $PW'_{VE-i}$, and biometrics $Bio'_{VE-i}$. Then, $VE_i$ computes $Rep(Bio'_{VE-i}, hs_{VE-i}) = eb'_{VE-i}$ using fuzzy extractor, $ns'_{VE-i} = zns_{VE-i} \oplus h(ID'_{VE-i} \| eb'_{VE-i})$, $PID'_{VE-i} = zPID_{VE-i} \oplus h(ns'_{VE-i} \| eb'_{VE-i} \| PW'_{VE-i})$, $MID'_{VE-i} = h(ns'_{VE-i} \| ID'_{VE-i} \| eb'_{VE-i})$, $CMK'_{TA-Vi} = zCMK_{TA-Vi} \oplus h(MID'_{VE-i} \| PID'_{VE-i} \| ID'_{VE-i})$, $sk'_{VE-i} = h(CMK'_{TA-Vi} \| eb'_{VE-i})$ $Pub'_{VE-i} = zPub_{VE-i} \oplus h(sk'_{VE-i} \| eb'_{VE-i} \| ns'_{VE-i})$ and $V'_{VE-i} = h(sk'_{VE-i} \| Pub'_{VE-i} \| ns'_{VE-i} \| PID'_{VE-i} \| MID'_{VE-i})$.

If $V'_{VE-i}$ is equal to $V_{VE-i}$, $VE_i$ picks a random nonce $rs_i$ and timestamp $ts_i$. Then, $VE_i$ computes $TP_{mv-i} = rs_i \cdot P$, $TK_{mv-i} = rs_i \cdot Pub_{ED-k}$, $VS_{mv-a} = PID_{VE-i} \oplus h(ts_i \parallel TK_{mv-i})$, $VS_{mv-b} = h(Pub_{VE-i} \parallel TK_{mv-i} \parallel PID_{VE-i} \parallel ts_i)$, and sends $\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}$ to the edge node $E_k$ through a public channel.

**LA2:** $E_k$ checks the freshness of $ts_i$ through the inequality $|ts_i - ts_c| < \Delta t$. Then, $E_k$ computes $PUF(psk_{ED-k}) = usk_{ED-k}$, $Rep(usk_{ED-k}, hs_{ED-k}) = eusk_{ED-k}$, $sk_{ED-k} = h(eusk_{ED-k} \parallel ID_{ED-k})$, $TK'_{mv-i} = TP_{mv-i} \cdot sk_{ED-k}$, $PID'_{VE-i} = VS_{mv-a} \oplus h(ts_i \parallel TK'_{mv-i})$, and retrieves $Pub_{VE-i}$. From that, $E_k$ checks the equality of $VS_{mv-b}$ and $h(Pub'_{VE-i} \parallel TK'_{mv-i} \parallel PID'_{VE-i} \parallel ts_i)$. If it is valid, $E_k$ generates $rs_k$ and $ts_k$, and computes $TP_{me-k} = rs_k \cdot P$, $TK_{me-k} = rs_k \cdot Pub_{VE-i}$, $SK_{Vi-Ek} = h(TK_{me-k} \parallel ts_i \parallel ts_k \parallel PID_{VE-i} \parallel Pub_{VE-i})$, and $ES_{me-a} = h(SK_{Vi-Ek} \parallel TK_{me-k} \parallel PID_{VE-i} \parallel Pub_{VE-i})$. $E_k$ sends $\{TP_{me-k}, ES_{me-a}, ts_k\}$ to $VE_i$ through a public channel.

**LA3:** $VE_i$ first check $|ts_k - ts_c| < \Delta t$ and computes $TK'_{me-k} = sk_{VE-i} \cdot TP_{me-k}$, $SK_{Ek-Vi} = h(TK'_{me-k} \parallel ts_i \parallel ts_k \parallel PID_{VE-i} \parallel Pub_{VE-i})$, and $ES'_{me-a} = h(SK_{Ek-Vi} \parallel TK_{me-k} \parallel PID_{VE-i} \parallel Pub_{VE-i})$. If $ES_{me-a}$ is equal to $ES'_{me-a}$, the session key $SK_{EK-Vi}$ is completely established between $VE_i$ and $E_k$.

### *4.4. Differential Privacy-Based Data Collection Phase*

After establishing the session key, the vehicle receives various edge intelligence services. To continuously improve and update the AI model of the edge node, the vehicle transmits some of the surrounding and personal information to the edge node. To achieve privacy protection and data anonymization, the proposed scheme securely utilizes user information based on local differential privacy. The detailed process is as follows.

**DC1:** With the collected data $cs_{VE-i}$, $VE_i$ executes Laplace mechanism $M(D) = f(D) + Lap(\Delta f/\epsilon)$ ($Lap(s_{ik}|\lambda) = \frac{1}{2\lambda} e^{-\frac{|s_{ik}|}{\lambda}}$, $s_i = [s_{i1}, \ldots, s_{ik}, \ldots, s_{in}]$) and obtains DP-based data $dp_{VE-i}$. After that, $VE_i$ generates a timestamp $ts_{Vi-dp}$ and computes $VE_{dc-a} = h(dp_{VE-i} \parallel ts_{Vi-dp} \parallel SK_{Ek-Vi})$, $VE_{dc-b} = dp_{VE-i} \oplus h(SK_{Ek-Vi} \parallel ts_{Vi-dp})$. $VE_i$ sends $\{VE_{dc-a}, VE_{dc-b}, ts_{Vi-dp}\}$ to $E_k$ via an wireless open channel.

**DC2:** $E_k$ checks the validity of $ts_{Vi-dp}$ and computes $dp'_{VE-i} = VE_{dc-b} \oplus h(SK_{Ek-Vi} \parallel ts_{Vi-dp})$, $VE'_{dc-a} = h(dp'_{VE-i} \parallel ts_{Vi-dp} \parallel SK_{Ek-Vi})$. If $VE'_{dc-a}$ is equal to $VE_{dc-a}$, $E_k$ utilizes the DP-based data $dp_{VE-i}$ for various service improvement tasks.

## 5. Security Analysis

In this section, we verify the security robustness of the proposed protocol using various methods of analysis, such as the ROR model, the Scyther tool, and informal security analysis.

### *5.1. ROR Model*

In various authentication protocols, each entity checks the legitimacy of the network partner and computes a session key. To verify the security of the session key, we use the ROR model [9]. We validate the security of the session key through various passive and active attacks of an adversary. Thus, the adversary conducts several games under the instantiated networks and attempts to distinguish random nonces and session keys using the test query. Thus, we define participants, adversaries, and queries to analyze the session key security of the proposed scheme using the ROR model. In the proposed scheme, four participants organize the system model: TA ($PM_{TA}^{a_1}$), cloud server ($PM_{CS}^{a_2}$), edge node ($PM_{EN}^{a_3}$), vehicle ($PM_{V}^{a_4}$). Note that $a_1$, $a_2$, $a_3$ and $a_4$ are the instance for the participants. The adversary has the ability to intercept, delete, and eavesdrop on messages through public channels. With this ability, the adversary can conduct various queries as follows:

- $E(PM_{TA}^{a_1}, PM_{CS}^{a_2}, PM_{EN}^{a_3}), PM_{V}^{a_4}$: The adversary can collect messages transmitted through public channels using $E(.)$ query.
- $C(PM_{V}^{a_4})$: The adversary can capture the vehicle and extract secret parameters using $C(.)$ query.

- $S(PM^a)$: This query represents a send event. Thus, the adversary can send messages to participant $PM^a$.
- $T(PM^a)$: This is a test query to distinguish the session key and random number. If the query $T(.)$ is executed, an unbiased coin is flipped. When the adversary obtains 0, the session key security can be achieved. However, the session key is not secure if the adversary obtains 1. Otherwise, the $NULL$ value is output.

Security Proof

**Theorem 1.** *We denote $CDS_{BR}(M)$ as the likelihood that an adversary cracks the security of the proposed scheme in polynomial time. We also define the total number of hash, send, and PUF queries as $tn_h$ and $tn_s$, and $tn_{PUF}$. The range space of the hash and the PUF function are denoted as $h(.)$ and $PUF(.)$. The Zipf's parameters [33] are $C'$ and $s'$. The probability of breaking the elliptic curve decisional Diffie–Hellman (ECDDH) problem and the number of bits in biometric parameters are defined as $CDS_{BR}^{ECC}(M)$ and $i_B$. Therefore, the proposed protocol can be secure when $CDS_{BR}(M)$ is less than the sum of that previously mentioned:*

$$CDS_{BR}(M) \leq \frac{tn_h^2}{|h|} + \frac{tn_{PUF}^2}{|PUF|} + 2CDS_{BR}^{ECC}(M) + 2\{Ctn_s^{s'}, \frac{tn_s}{2^{i_B}}\}$$

**Proof.** According to [34–36], we conduct six games ($G_k$, $k = 0, 1, 2, 3, 4, 5$). The advantage and winning probability of the adversary in each game as $A[WIN_{G_k}]$ and $WIN_{G_k}$.

$G_0$: In this game, the adversary does not have any information for the session key. Thus, the adversary selects a random bit $O$. By the definition in [9], we can obtain the following Equation (1):

$$CDS_{BR}(M) = |2A[WIN_{G_0}] - 1| \tag{1}$$

$G_1$: The adversary executes the $Exec(.)$ query and obtains $\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}$ and $\{TP_{me-k}, ES_{me-a}, ts_k\}$. Then, the adversary conducts the $T(.)$ query to verify whether the session key is secure or not. However, the adversary cannot decrypt messages because each parameter utilized various forms of security technology, such as ECC, PUF, and biometrics, in the proposed scheme. This means that the adversary has the same probability of winning the game as $G_0$. Thus, the winning possibility is same as $A[WIN_{G_0}]$. We can obtain the following Equation (2):

$$A[WIN_{G_0}] = A[WIN_{G_1}] \tag{2}$$

$G_2$: Using the send and hash queries, the adversary tries to reveal the session key security in this game. However, the proposed protocol can resist hash-collision problems through the use of the "cryptographic one-way hash function". Thus, we can obtain the following inequality (3) using the birthday paradox [37]:

$$A[WIN_{G_2}] - A[WIN_{G_1}] \leq \frac{tn_h^2}{|h|} \tag{3}$$

$G_3$: The adversary utilizes send and $PUF$ queries to break the security of the session key. According to Section 3.4, it is practically impossible to guess the secret parameter derived from PUF circuit, which means that the adversary cannot reveal the secret key of edge nodes. Thus, we can obtain the inequality (4), which is similar to (3):

$$|A[WIN_{G_3}] - A[WIN_{G_2}]| \leq \frac{tn_{PUF}^2}{|PUF|} \tag{4}$$

$G_4$: The adversary tries to compute the session key using $\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}$ and $\{TP_{me-k}, ES_{me-a}, ts_k\}$. However, $TK_{mv-i} = rs_i \cdot sk_{ED-k} \cdot P$ and $TK_{me-k} = rs_k \cdot sk_{VE-i} \cdot$

$P$ have security based on the ECDDH problem. Thus, the winning probability of $G_4$ is solving this problem in polynomial time. The inequality (9) can be obtained:

$$|A[WIN_{G_4}] - A[WIN_{G_3}]| \leq CDS_{BR}^{ECC}(M) \tag{5}$$

$G_5$: This game is the final game in which the adversary collects the secret parameter of the vehicle using $C(.)$ query. After that, the adversary tries to compute the secret parameters using $\{zns_{VE-i}, zPID_{VE-i}, zCMK_{TA-Vi}, zPub_{VE-i}, V_{VE-i}, hs_{VE-i}\}$. However, the proposed scheme utilizes the identity, password, and biometrics to perform local login process. Thus, it is a computationally infeasible task to guess them simultaneously. Therefore, we can obtain the inequality (6) using Zipf's parameters:

$$|A[WIN_{G_5}] - A[WIN_{G_4}]| \leq \{Ctn_s^{s'}, \frac{tn_s}{2^{i_B}}\} \tag{6}$$

After $G_5$, the adversary guesses a bit $t$. Because the winning probability in $G_5$ is 0.5, we can obtain the Equation (7):

$$A[WIN_{G_5}] = \frac{1}{2} \tag{7}$$

We can obtain the following after uniting Equations (1) and (2):

$$\begin{aligned} \frac{1}{2}CDS_{BR}(M) &= |A[WIN_{G_0}] - \frac{1}{2}| \\ &= |A[WIN_{G_1}] - \frac{1}{2}| \end{aligned} \tag{8}$$

We also obtain the following after uniting Equations (7) and (8):

$$\frac{1}{2}CDS_{BR}(M) = |A[WIN_{G_1}] - A[WIN_{G_5}]| \tag{9}$$

We obtain the following after using (9) and triangular inequality:

$$\begin{aligned} &\frac{1}{2}CDS_{BR}(M) = |A[WIN_{G_1}] - A[WIN_{G_5}]| \\ &\leq |A[WIN_{G_1}] - A[WIN_{G_4}]| + |A[WIN_{G_4}] - A[WIN_{G_5}]| \\ &\leq |A[WIN_{G_1}] - A[WIN_{G_2}]| + |A[WIN_{G_2}] - A[WIN_{G_3}]| \\ &+ |A[WIN_{G_3}] - A[WIN_{G_4}]| + |A[WIN_{G_4}] - A[WIN_{G_5}]|. \\ &\leq \frac{tn_h^2}{2|h|} + \frac{tn_{PUF}^2}{2|PUF|} + CDS_{BR}^{ECC}(M) + \{Ctn_s^{s'}, \frac{tn_s}{2^{i_B}}\} \end{aligned} \tag{10}$$

After multiplying (10) by 2, we can obtain the following result, which is same as Theorem 1:

$$CDS_{BR}(M) \leq \frac{tn_h^2}{|h|} + \frac{tn_{PUF}^2}{|PUF|} + 2CDS_{BR}^{ECC}(M) + 2\{Ctn_s^{s'}, \frac{tn_s}{2^{i_B}}\}$$

□

### *5.2. Informal Analysis*

#### 5.2.1. Replay and Man-in-the-Middle Attacks

The adversary can capture messages from the public channel and send them to other network participants. In the proposed login and authentication phase, each entity generates and sends timestamp $ts$ to prove the freshness of message. If the timestamp is out of time, the communication partner regards the message as failed information. Thus, the adversary cannot have an advantage when using replay and man-in-the-middle attacks.

5.2.2. Impersonation Attacks

In this attack, the adversary attempts to disguise itself as a legitimate user using messages transmitted via an open channel. Thus, the adversary must generate $TP_{mv-i}$, $VS_{mv-a}$, $VS_{mv-b}$, and $ts_i$, which are the elements of authentication request message. However, the adversary cannot generate $VS_{mv-a}$ because $PID_{VE-i}$ is a secret parameter of the legitimate vehicle $VE_i$. Thus, the adversary cannot compute the message. For the reason above, the proposed scheme can prevent impersonation attacks.

5.2.3. Insider Attacks

In this attack, an adversary registers with the TA as a vehicle and performs the login and authentication phase. Then, the adversary collects public messages to reveal secret credentials. With the leaked credentials, the adversary invades the other vehicle's session and tries to compute the session key. However, the adversary cannot decrypt any sensitive information because of the use of ECC and PUF. To compute $PID_{VE-i}$ and $SK_{Ek-Vi}$, the adversary must obtain $TK_{mv-i}$ and $TK_{me-k}$, which are based on the ECC and PUF technology. Therefore, the proposed scheme has robustness against insider attacks.

5.2.4. Privileged Insider Attacks

In the real environment, users utilize same identity and password in various network systems. Thus, a privileged insider attempts to compute the identity and password of legitimate users in this attack. In the registration phase, the adversary can obtain the identity $ID_{VE-i}$. However, the adversary cannot guess the password of $VE_i$ because $\{zns_{VE-i}, zPID_{VE-i}, zCMK_{TA-Vi}, zPub_{VE-i}, V_{VE-i}, hs_{VE-i}\}$ are masked in biometrics $Bio_{VE-i}$. Thus, the proposed protocol can prevent privileged insider attacks.

5.2.5. Verification Table Leakage Attacks

In this attack, the adversary obtains the verification table $\{ID_{ED-k}, h(ID_{ED-K} \parallel ns_{ED-k})\}$ and $\{PID_{VE-i}, SID_{VE-i}, ns_{TA-Vi}\}$. From this information, the adversary can try to compute the session key $SK_{Ek-Vi} = h(TK_{me-k} \parallel ts_i \parallel ts_k \parallel PID_{VE-i} \parallel Pub_{VE-i})$. However, the adversary cannot compute the session key because $TK_{me-k}$ is composed of $sk_{VE-i}$, which is the secret key of $VE_i$. Thus, the proposed scheme is secure against verification table leakage attacks.

5.2.6. Ephemeral Secret Leakage (ESL) Attacks

In this attack, an adversary tries to compute the session key if the ephemeral secret parameters $rs_i$ and $rs_k$ are leaked. To compute the session key, the adversary must obtain $TK_{me-k}$, $PID_{VE-i}$, and $Pub_{VE-i}$. However, the adversary still does not have the secret key $sk_{VE-i}$ which means the adversary cannot compute $TK_{me-k}$. Thus, the proposed scheme can prevent ESL attacks.

5.2.7. Perfect Forward Secrecy

If an adversary obtains the master key $mk_{TA}$ of TA, it can try to leak the secret parameters. However, the adversary has no advantage from that because all messages are masked in ECC and the secret parameter $sk_{VE-i}$. Thus, the proposed protocol can achieve perfect forward secrecy.

5.2.8. User Anonymity and Untraceability

In edge intelligence-enabled VANET environments, the history of a vehicle can be critical information. Thus, the anonymity and untraceability must be protected in the proposed scheme. In the proposed protocol, $VE_i$ sends a temporal parameter $VS_{mv-a}$ to guarantee freshness and the confusion of identity. Thus, the adversary cannot specify the actual vehicle from the message. Thus, the proposed protocol can achieve anonymity and untraceability.

#### 5.2.9. Mutual Authentication

When the vehicle $VE_i$ tries to authenticate with the edge node, $VE_i$ generates a request message using secret parameters and a random number and timestamp. The edge node checks the freshness of the timestamp using $\Delta t$ and verifies the legitimacy of $VE_i$ using ECC and PUF. If the process is a success, the edge node can demonstrate that $VE_i$ is a legitimate participant. Thus, the proposed protocol can guarantee mutual authentication.

### *5.3. Scyther Tool*

We evaluate the security of the proposed protocol using an automatic verification and simulation tool, named Scyther [10,11]. The Scyther tool analyzes possible behavior patterns in security protocols and evaluates various security properties, such as the robustness of the authentication and the confidentiality of variables. The Scyther tool can represent the behaviors of the security protocol by characterizing protocols. Thus, we convert the proposed scheme into SPDL (Security Protocol Description Language), which is the programming language used in the Scyther tool. Then, the Scyther tool conducts a security simulation. After that, the Scyther tool conducts the security verification using various claim events, which are described in Table 3. When the protocol is secure and well-authenticated, the Scyther tool outputs "OK" and "No attacks" in the results window. Figure 5 shows that the proposed scheme is secure against various security attacks and has robust mutual authentication.

**Table 3.** Claim events in Scyther tool.

| Claim Event | Description |
|---|---|
| Aliveness | The entity is certain whether or not it is communicating with the other party. |
| Weak agreement | The entity is certain whether or not it is communicating with the other legitimate party. |
| Non-injective agreement | The entity is certain whether or not it is communicating with the other legitimate party, which exchanges the legal data. |
| Non-injective synchronization | The entity is certain whether or not it is communicating with the other legitimate party, which exchanges the legal data. Moreover, the messages are transmitted, following the rules of the protocol. |

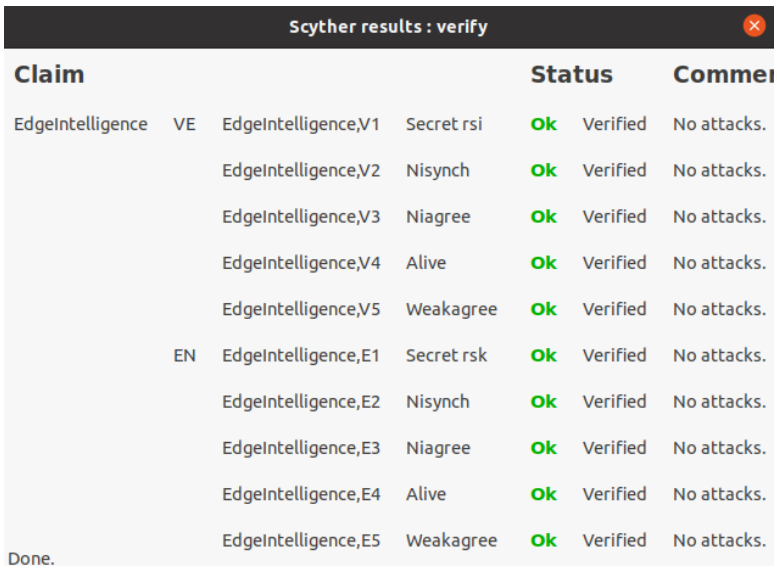


**Figure 5.** Results window of the proposed scheme using the Scyther tool.

## 6. Performance Analysis

In this section, we measure the computational, communicational overhead of the proposed protocol. Based on the results, we conduct comparative studies with the related schemes. Moreover, we simulate the practical deployment of the proposed scheme using NS-3.

### *6.1. MIRACL Testbed*

MIRACL [13] is a C/C++ language-based open-source SDK that can implement various security schemes using built-in cryptographic primitives. MIRACL can be effectively applied to small equipment such as embedded and mobile devices through the optimization of cryptographic primitives. In our paper, we measure ECC multiplication ($C_{E-mul}$), ECC addition ($C_{E-add}$), AES encryption ($C_{A-enc}$), AES decryption ($C_{A-dec}$), bilinear pairings ($C_{BP}$), exponentiation ($C_{exp}$), and hash function ($C_{Hash}$) using MIRACL. The testbed environments in our study are as follows:

- Desktop environments: "Linux Ubuntu 20.04 LTS, Intel Core i3-8100 CPU @ 3.60 GHz, 16 GB RAM"
- Raspberry Pi environments: Raspberry Pi 4B (Quad-core ARM Cortex-A72 @ 1.5 GHz, 8 GB RAM)

We conduct the experimental study using these environments to measure the execution time for each of the cryptographic primitives. We execute the cryptographic primitives for 100 times and deduce the results. Tables 4 and 5 represent the maximum, minimum, and average execution times for each cryptographic primitive.

**Table 4.** MIRACL testbed result of Raspberry Pi platform.

| Notations | Max | Min | Average |
|---|---|---|---|
| ECC multiplication ($C_{E-mul}$) | 3.018 ms | 2.003 ms | 2.265 ms |
| ECC addition ($C_{E-add}$) | 0.035 ms | 0.018 ms | 0.024 ms |
| AES encryption ($C_{A-enc}$) | 0.007 ms | 0.003 ms | 0.004 ms |
| AES decryption ($C_{A-dec}$) | 0.006 ms | 0.003 ms | 0.004 ms |
| Bilinear pairings ($C_{BP}$) | 13.837 ms | 10.533 ms | 11.937 ms |
| Exponentiation ($C_{exp}$) | 0.189 ms | 0.093 ms | 0.115 ms |
| Hash function ($C_{Hash}$) | 0.007 ms | 0.005 ms | 0.006 ms |

**Table 5.** MIRACL testbed result of desktop platform.

| Notations | Max | Min | Average |
|---|---|---|---|
| ECC multiplication ($C_{E-mul}$) | 0.421 ms | 0.388 ms | 0.409 ms |
| ECC addition ($C_{E-add}$) | 0.006 ms | 0.002 ms | 0.006 ms |
| AES encryption ($C_{A-enc}$) | 0.001 ms | 0.001 ms | 0.001 ms |
| AES decryption ($C_{A-dec}$) | 0.001 ms | 0.001 ms | 0.001 ms |
| Bilinear pairings ($C_{BP}$) | 2.735 ms | 2.015 ms | 2.253 ms |
| Exponentiation ($C_{exp}$) | 0.052 ms | 0.033 ms | 0.041 ms |
| Hash function ($C_{Hash}$) | 0.002 ms | 0.001 ms | 0.002 ms |

### 6.2. Computational Overheads

In this section, we conduct a comparative study of our proposed scheme with other related research [22–27] in terms of computational overhead. In the proposed login and authentication phase, the vehicle uses 3 ECC multiplications ($C_{E-mul}$) and 11 hash functions ($C_{Hash}$). Additionally, the edge node performs 3 ECC multiplications and 5 hash functions. Based on Tables 4 and 5, we measure the computational overhead of the vehicle and the edge node. The overall overhead is shown in Table 6. The proposed scheme has from 10% to 48% better performance compared with the state-of-the-art research [26,27]. Therefore, the results demonstrate that the proposed scheme uses lower computational overheads compared to other related schemes [22–27].

**Table 6.** Comparative study of computational overheads.

| Schemes | Device | Infrastructure | Total Costs |
|---|---|---|---|
| Jia et al. [22] | $5C_{E-mul} + C_{E-add} + 4C_{Hash}$ | $5C_{E-mul} + 2C_{E-add} + 5C_{Hash} + C_{BP}$ | 15.693 ms |
| Bagga et al. [23] | $5C_{E-mul} + C_{E-add} + 8C_{Hash}$ | $4E_{E-mul} + C_{E-add} + 7C_{Hash}$ | 13.053 ms |
| Ke et al. [24] | $4C_{exp} + 2C_{BP}$ | $4C_{BP}$ | 33.346 ms |
| Seifelnasr et al. [25] | $10C_{E-mul} + 2C_{E-add} + 3C_{Hash}$ | $9C_{E-mul} + 5C_{E-add} + 2C_{Hash}$ | 26.431 ms |
| Yadav et al. [26] | $3C_{E-mul} + 2C_{E-add} + 5C_{Hash}$ | $5C_{E-mul} + 2C_{E-add} + 4C_{Hash}$ | 8.938 ms |
| Kumar and Om [27] | $4C_{E-mul} + 7C_{Hash} + C_{A-enc}$ | $7C_{E-mul} + 11C_{Hash} + 2C_{A-enc} + 3C_{A-dec}$ | 11.367 ms |
| Proposed | $3C_{E-mul} + 11C_{Hash}$ | $3C_{E-mul} + 5C_{Hash}$ | 8.098 ms |

We also analyze computation complexity through the primitives used in the proposed scheme. In the initialization phase, TA picks various random numbers and selects an elliptic curve. Thus, the computation complexity is $O(n^2)$. In the registration phase, edge nodes and vehicles register with the network using the hash function, PUF, and fuzzy extractor, which can be indicated as $O(k)$ and $O(1)$. In the login and authentication phase, vehicle and edge nodes utilize various ECC multiplication, hash function and exclusive-OR computations. Thus, computation complexity is $O(n^2)$, $O(k)$, and $O(n)$ in this phase. Through the analysis conducted across the proposed scheme, the computation complexity is $O(n^2)$.

### 6.3. Communication Overheads

We analyze the communication overhead of the proposed scheme to verify the efficiency. To measure the message load on the public channel during the authentication phase, we define the communication cost as follows: the ECC point, hash, random number, identity, and timestamp are 320, 160, 160, 160, and 32, respectively. Thus, the messages in the proposed method are $\{TP_{mv-i}, VS_{mv-a}, VS_{mv-b}, ts_i\}$ and $\{TP_{me-k}, ES_{me-a}, ts_k\}$, amounting to $(320 + 160 + 160 + 32) + (320 + 160 + 32) = 1184$ bits. Table 7 presents the results of measuring the overall communication overhead and the number of messages for the proposed scheme and other related schemes [22–26]. The results show that the proposed scheme has lower communication overhead than [22,23,25–27], and slightly higher than [24]. However, the proposed scheme has lower computational overhead compared to the comparison schemes [22–27].

**Table 7.** Comparative study of communicational overheads.

| Schemes | Total Communication Costs | Messages |
|---|---|---|
| Jia et al. [22] | 1504 bits | 2 |
| Bagga et al. [23] | 1856 bits | 3 |
| Ke et al. [24] | 992 bits | 2 |
| Seifelnasr et al. [25] | 3840 bits | 3 |
| Yadav et al. [26] | 1472 bits | 3 |
| Kumar and Om [27] | 2880 bits | 5 |
| Proposed | 1184 bits | 2 |

### *6.4. NS-3 Simulation*

In this section, we conducted a simulation study to estimate the practical deployment of the proposed scheme using NS-3 [12]. In NS-3, each network node is executed according to the coded application layer. Then, the node generates a network packet to "NetDevice" and sends it to the other node through a "Channel". The proposed scheme is composed of several edge nodes and various vehicles in the mutual authentication phase. In our system model, edge nodes are fixed infrastructures to communicate with vehicles that have dynamic movement properties. Moreover, message bytes are 84 and 64 bytes in our login and authentication phase. We conducted NS-3 simulation under desktop platform (Intel(R) Core(TM) i5-11400 @ 2.60 GHz with 24.0 GB RAM, Ubuntu 16.04 LTS). Table 8 represents the parameters used in our NS-3 simulation study. With these parameters, we simulate the proposed scheme using NS-3 through four scenarios as follows:

- *Scenario 1:* 10 vehicles are placed in a single edge node's service range.
- *Scenario 2:* 30 vehicles are placed in a single edge node's service range.
- *Scenario 3:* 60 vehicles are placed in three edge nodes' service range.
- *Scenario 4:* 90 vehicles are placed in three edge nodes' service range.

**Table 8.** NS-3 parameters in our simulation.

| **Simulation Parameters** | **Details** |
|---|---|
| Version of NS-3 | 3.29 |
| Version of OS | Ubuntu 16.04 LTS |
| Number of vehicles | 10, 30, 60, 90 |
| Number of edge nodes | 1, 3 |
| Propagation loss model | TwoRayGroundPropagationLossModel |
| Mobility model | RandomDirection2dMobilityModel<br>ConstantPositionMobilityModel |
| Simulation area | 500 m × 500 m |
| Wireless channel bandwidth | 6 Mbps |
| Network | IEEE 802.11p |
| Routing protocol | Ad hoc On-demand Distance Vector |
| Simulation time | 300 s |

Throughput and End-to-End Delay Analysis

We perform throughput analysis to determine whether the proposed scheme can provide VANET services. Since throughput is the minimum transmission capability including protocol efficiency in the end-to-end data path, we can measure the performance of the proposed scheme. We define $Pa_{recv}$, $Si_{packet}$, $Ti_{tot}$ as the number of received packets, the packet size, and the total time, respectively. Thus, the formula of throughput is as follows:

$$\frac{Pa_{recv} \times |Si_{packet}|}{Ti_{tot}}$$

Also, we measure the end-to-end delay recording the time it takes data to move from one point to another. We define total packets, and one data packet; the times for receiving and sending messages are $Pa_{tot}$, $k$, $Ti_{recv}$, and $Ti_{send}$, respectively. Thus, the formula of the end-to-end delay is as follows:

$$\frac{\Sigma_{k=1}^{Pa_{tot}} (Ti_{recv} - Ti_{send})}{Pa_{tot}}$$

Through the throughput and end-to-end delay formulas, we conduct the NS-3 simulation study according to the four scenarios. The results are shown in Figure 6.

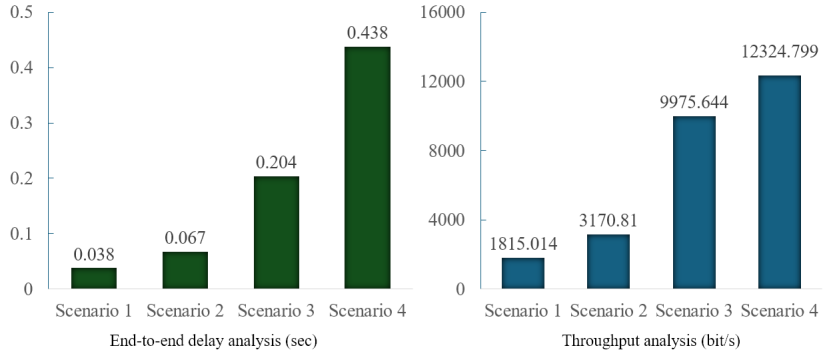


**Figure 6.** Results of the end-to-end delay and throughput analyses using NS-3.

### *6.5. Security Features*

We show the security and functionality features of the proposed scheme and the related schemes [22–27] in Table 9. According to Table 9, the proposed scheme can prevent various security attacks, including replay, impersonation, verification table leakage, ESL, and insider attacks. Moreover, the proposed scheme can ensure anonymity and perfect forward secrecy. Thus, we can demonstrate that the proposed scheme has high security and functionality features compared with the related schemes [22–27].

**Table 9.** Comparison of security and functionality features.

| Security Features | [22] | [23] | [24] | [25] | [26] | [27] | Proposed |
|---|---|---|---|---|---|---|---|
| F1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| F2 | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| F3 | ○ | × | ○ | ○ | ○ | ○ | ○ |
| F4 | — | — | — | — | — | ○ | ○ |
| F5 | ○ | ○ | ○ | ○ | — | — | ○ |
| F6 | ○ | ○ | ○ | ○ | — | — | ○ |
| F7 | × | ○ | ○ | ○ | ○ | ○ | ○ |
| F8 | × | ○ | ○ | ○ | ○ | ○ | ○ |
| F9 | ○ | ○ | ○ | × | ○ | ○ | ○ |
| F10 | × | ○ | × | × | ○ | ○ | ○ |
| F11 | × | × | ○ | × | × | × | ○ |
| F12 | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

○: "Provides the security and functionality features"; ×: "Does not provide the security and functionality features"; —: "Does not consider features". Note: (F(Feature)1: Replay attacks), (F2: Man-in-the-middle attacks), (F3: Impersonation attacks), (F4: Insider attacks), (F5: Privileged insider attacks), (F6: Verification table leakage attacks), (F7: ESL attacks), (F8: Perfect forward secrecy), (F9: Anonymity and untraceability), (F10: High computation overhead), (F11: High communication overhead), (F12: Formal analysis).

## 7. Conclusions

In this paper, we proposed a secure authentication scheme for edge intelligence-enabled VANET environments. The proposed scheme can provide secure and efficient mutual authentication between edge nodes and vehicles using PUF, biometrics, and ECC. With the established session key, vehicles can receive various edge intelligence services. Moreover, the proposed scheme can support a privacy-preserving data collection scheme using local differential privacy. We conducted various security analyses, including the use of the ROR model, the Scyther tool, and carrying out an informal security analysis, to prove the security robustness of the proposed protocol. Furthermore, we measured the performance of cryptographic primitives using MIRACL SDK under Raspberry Pi 4B and a desktop platform. Based on the performance result, we compare the computational and communication overheads of the proposed scheme with the related schemes. We simulated the proposed protocol to check the practical deployment in VANET environments using NS-3. In future work, we will extend the proposed scheme considering edge intelligence-enabled VANET environments. In addition, we will perform various analyses, such as the

scalability test, and a machine learning analysis using differential privacy-based actual VANET data.

**Author Contributions:** Conceptualization, D.K.; methodology, D.K. and K.P.; software, D.K. and S.S.; validation, S.S. and K.P.; formal analysis, D.K.; writing—original draft preparation, D.K.; writing—review and editing, S.S. and K.P.; supervision, Y.P.; project administration, Y.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education under grant 2020R1I1A3058605.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Liu, Y.; Peng, M.; Shou, G.; Chen, Y.; Chen, S. Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 6722–6747. [CrossRef]
2. Plastiras, G.; Terzi, M.; Kyrkou, C.; Theocharides, T. Edge intelligence: Challenges and opportunities of near-sensor machine learning applications. In Proceedings of the 2018 IEEE 29th International Conference on Application-Specific Systems, Architectures and Processors (ASAP), Milan, Italy, 10–12 July 2018; pp. 1–7.
3. Zhang, J.; Letaief, K.B. Mobile edge intelligence and computing for the internet of vehicles. *Proc. IEEE* **2019**, *108*, 246–261. [CrossRef]
4. Balasubramanian, V.; Otoum, S.; Reisslein, M. VeNet: Hybrid stacked autoencoder learning for cooperative edge intelligence in IoV. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16643–16653. [CrossRef]
5. Haris, M.; Shah, M.A.; Maple, C. Internet of intelligent vehicles (IoIV): An intelligent VANET based computing via predictive modeling. *IEEE Access* **2023**, *11*, 49665–49674. [CrossRef]
6. Dwork, C. Differential privacy. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Venice, Italy, 10–14 July 2006; pp. 1–12.
7. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology-EUROCRYPT 2004, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
8. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
9. Abdalla, M.; Fouque, P.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In *Public Key Cryptography—PKC 2005, Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005*; Lecture Notes in Computer Science (LNCS); Springer: Berlin/Heidelberg, Germany, 2005; pp. 65–84.
10. Cremers, C.J. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols: Tool Paper. In Proceedings of the International Conference on Computer Aided Verification, Princeton, NJ, USA, 7–14 July 2008; pp. 414–418.
11. Scyther Tool. Available online: https://people.cispa.io/cas.cremers/scyther/ (accessed on 5 July 2024).
12. NS-3.29. Available online: https://www.nsnam.org (accessed on 5 July 2024)
13. MIRACL Cryptographic SDK. Available online: https://github.com/miracl/MIRACL (accessed on 5 July 2024).
14. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* **2019**, *107*, 1738–1762. [CrossRef]
15. Deng, S.; Zhao, H.; Fang, W.; Yin, J.; Dustdar, S.; Zomaya, A.Y. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet Things J.* **2020**, *7*, 7457–7469. [CrossRef]
16. Qi, W.; Li, Q.; Song, Q.; Guo, L.; Jamalipour, A. Extensive edge intelligence for future vehicular networks in 6G. *IEEE Wirel. Commun.* **2021**, *28*, 128–135. [CrossRef]
17. Gong, T.; Zhu, L.; Yu, F.R.; Tang, T. Edge intelligence in intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 8919–8944. [CrossRef]
18. Zhang, Y.; Huang, H.; Yang, L.X.; Xiang, Y.; Li, M. Serious challenges and potential solutions for the industrial internet of things with edge intelligence. *IEEE Netw.* **2019**, *33*, 41–45. [CrossRef]
19. Li, Y.; Yu, Y.; Susilo, W.; Hong, Z.; Guizani, M. Security and privacy for edge intelligence in 5G and beyond networks: Challenges and solutions. *IEEE Wirel. Commun.* **2021**, *28*, 63–69. [CrossRef]
20. Xu, D.; Li, T.; Li, Y.; Su, X.; Tarkoma, S.; Jiang, T.; Crowcroft, J.; Hui, P. Edge intelligence: Empowering intelligence to the edge of network. *Proc. IEEE* **2021**, *109*, 1778–1837. [CrossRef]
21. Villar-Rodriguez, E.; Pérez, M.A.; Torre-Bastida, A.I.; Senderos, C.R.; López-de-Armentia, J. Edge intelligence secure frameworks: Current state and future challenges. *Comput. Secur.* **2023**, *130*, 103278. [CrossRef]
22. Jia, X.; He, D.; Kumar, N.; Choo, K.K.R. A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. *IEEE Syst. J.* **2019**, *14*, 560–571. [CrossRef]

23. Bagga, P.; Das, A.K.; Wazid, M.; Rodrigues, J.J.; Choo, K.K.R.; Park, Y. On the design of mutual authentication and key agreement protocol in internet of vehicles-enabled intelligent transportation system. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1736–1751. [CrossRef]
24. Ke, C.; Zhu, Z.; Xiao, F.; Huang, Z.; Meng, Y. SDN-based privacy and functional authentication scheme for fog nodes of smart healthcare. *IEEE Internet Things J.* **2022**, *9*, 17989–18001. [CrossRef]
25. Seifelnasr, M.; AlTawy, R.; Youssef, A.; Ghadafi, E. Privacy-preserving mutual authentication protocol with forward secrecy for IoT-edge-cloud. *IEEE Internet Things J.* **2023**, *11*, 8105–8117. [CrossRef]
26. Yadav, A.K.; Shojofar, M.; Braeken, A. iVFAS: An improved vehicle-to-fog authentication system for secure and efficient fog-based road condition monitoring. *IEEE Trans. Veh. Technol.* **2024**, 1–16. [CrossRef]
27. Kumar, P.; Om, H. Multi-TA model-based conditional privacy-preserving authentication protocol for fog-enabled VANET. *Veh. Commun.* **2024**, *47*, 100785. [CrossRef]
28. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
29. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In *Advances in Cryptology—EUROCRYPT 2002, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, 28 April–2 May 2002*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 337–351.
30. Oh, J.; Son, S.; Kwon, D.; Kim, M.; Park, Y.; Park, Y. Design of secure and privacy-preserving data sharing scheme based on key aggregation and private set intersection in medical information system. *Mathematics* **2024**, *12*, 1717. [CrossRef]
31. Son, S.; Oh, J.; Kwon, D.; Kim, M.; Park, K.; Park, Y. A Privacy-preserving authentication scheme for a blockchain-based energy trading system. *Mathematics* **2023**, *11*, 4653. [CrossRef]
32. Hou, W.; Sun, Y.; Li, D.; Guan, Z.; Liu, J. Lightweight and privacy-preserving charging reservation authentication protocol for 5G-V2G. *IEEE Trans. Veh. Technol.* **2023**, *72*, 7871–7883. [CrossRef]
33. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
34. Park, K.; Lee, J.; Das, A.K.; Park, Y. BPPS: Blockchain-enabled privacy-preserving scheme for demand-response management in smart grid environments. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 1719–1729. [CrossRef]
35. Park, K.; Park, Y. MIoT-CDPS: Complete decentralized privacy-preserving scheme for medical internet of things. *Internet Things* **2024**, *27*, 101250. [CrossRef]
36. Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A.K. Design of blockchain-based lightweight V2I handover authentication protocol for VANET. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1346–1358. [CrossRef]
37. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; pp. 156–171.