applied sciences

MDPI

Article

A Secure and Efficient Authentication Scheme for Fog-Based Vehicular Ad Hoc Networks

Sangjun Lee [1], Seunghwan Son [1], DeokKyu Kwon [1], Yohan Park [2] and Youngho Park [1,*]

[1] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea; gumoning9010@knu.ac.kr (S.L.); sonshawn@knu.ac.kr (S.S.); kdk145@knu.ac.kr (D.K.)
[2] School of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea; yhpark@kmu.ac.kr
* Correspondence: parkyh@knu.ac.kr

Abstract: Recently, the application of fog-computing technology to vehicular ad hoc networks (VANETs) has rapidly advanced. Despite these advancements, challenges remain in ensuring efficient communication and security. Specifically, there are issues such as the high communication and computation load of authentications and insecure communication over public channels between fog nodes and vehicles. To address these problems, a lightweight and secure authenticated key agreement protocol for confidential communication is proposed. However, we found that the protocol does not offer perfect forward secrecy and is vulnerable to several attacks, such as privileged insider, ephemeral secret leakage, and stolen smart card attacks. Furthermore, their protocol excessively uses elliptic curve cryptography (ECC), resulting in delays in VANET environments where authentication occurs frequently. Therefore, this paper proposes a novel authentication protocol that outperforms other related protocols regarding security and performance. The proposed protocol reduced the usage frequency of ECC primarily using hash and exclusive OR operations. We analyzed the proposed protocol using informal and formal methods, including the real-or-random (RoR) model, Burrows–Abadi–Nikoogadam (BAN) logic, and automated validation of internet security protocols and applications (AVISPA) simulation to show that the proposed protocol is correct and secure against various attacks. Moreover, We compared the computational cost, communication cost, and security features of the proposed protocol with other related protocols and show that the proposed methods have better performance and security than other schemes. As a result, the proposed scheme is more secure and efficient for fog-based VANETs.

Keywords: fog computing; vehicular ad hoc network; lightweight; key agreement; perfect forward secrecy; BAN logic; RoR model; AVISPA simulation

Academic Editors: Larysa Titarenko, Kazimierz Krzywicki and Alexander Barkalov

Received: 3 December 2024
Revised: 20 January 2025
Accepted: 23 January 2025
Published: 25 January 2025

Citation: Lee, S.; Son, S.; Kwon, D.; Park, Y.; Park, Y. A Secure and Efficient Authentication Scheme for Fog-Based Vehicular Ad Hoc Networks. *Appl. Sci.* **2025**, *15*, 1229. https://doi.org/10.3390/app15031229

1. Introduction

The vehicular ad hoc network (VANET) [1] offers a promising approach to improving communication and data sharing between vehicles and infrastructure, transforming contemporary transportation. The VANET facilitates several applications, including road safety support, modernized traffic management, and improved driving experiences [2]. To provide these services, it is essential to handle the large amounts of traffic data generated by vehicles, which requires rapid data transmission and real-time data processing [3]. Traditional VANET architecture has used cloud computing technology for data storage and processing to satisfy these requirements. However, the cloud server is far from the

vehicle; thus, processing the numerous data generated by a vehicle results in high latency and communication costs.

Fog computing [4] is a promising solution to improve the functionality of VANET environments, providing real-time processing and storage capabilities and using the communication and computing resources of each vehicle more efficiently. Furthermore, fog computing enables efficient data processing, enhanced scalability, and low-latency communication by extending the concept of cloud computing to the network edge in VANET environments [5]. Therefore, integrating fog computing with VANETs is necessary to improve the capabilities of autonomous vehicles.

However, fog-based VANETs encounter security challenges that threaten road safety and system integrity [6]. The interconnectedness of vehicles makes them vulnerable to cyberattacks, and messages transmitted over VANETs on public channels can be tampered with, replayed, intercepted, or deleted by an attacker. Moreover, the dynamic characteristics of fog nodes and reliance on wireless communication necessitate a robust authentication scheme to safeguard operations in fog-based VANETs [7]. Fog-based VANETs must satisfy certain security requirements, such as secure data transmission, privacy protection, and authentication.

Therefore, robust security protocols must be developed to authenticate entities and reduce potential threats. In 2024, Awais et al. [8] proposed a secure and lightweight authentication scheme to strengthen the security of fog-based VANETs. However, problems typically occur, such as session key exposure due to ephemeral secret leakage attacks and high communication costs due to the frequent use of public keys. Therefore, this paper proposes an improved protocol to address these security concerns effectively and enhance the overall reliability and efficiency of fog-based VANETs.

2. Related Works

This study introduces several papers that have described fog-based VANETs. Hou et al. [9] proposed a vehicular fog-computing architecture that uses vehicles as the infrastructures to improve communication and computational capacity. This architecture performs communication and computation by efficiently employing the resources of individual vehicles via a collaborative aggregation of end-user clients or nearby edge devices. Combining the resources of individual vehicles significantly improves the quality of vehicular applications and services. Peixoto et al. [10] proposed a framework for data clustering to reduce traffic data at the edge of vehicular networks using fog computing. The proposed framework for data clustering introduces two techniques to minimize the traffic information flow: a baseline technique that detects traffic congestion and two modified clustering techniques that order points to identify the clustering structure and density-based spatial clustering of applications with noise. This framework maintains high accuracy, even in highly congested vehicular traffic conditions, and reduces the communication costs in VANETs. Pereira et al. [11] introduced a framework for applying fog-computing technology in a VANET environment. Furthermore, they proposed a proof-of-concept system for data analyses in a fog-based VANET environment. Their study applied actual VANET data to demonstrate that fog computing is as effective as cloud computing. Their study demonstrated that distributed fog nodes can cooperate to process crucial data, quickly providing reliable data for smart city decision support systems. Farooqi et al. [12] designed a priority-based fog-computing model for smart-city vehicle transportation to reduce delays and latency. When the fog node was overloaded, they redirected high-priority requests to an adjacent node and transmitted low-priority requests to the cloud for additional processing. This technique reduced latency and delays by 20% and 35%, respectively, compared to the cloud computing architecture, allowing efficient communication between devices.

This research introduces several papers describing authentication protocols in wireless communication environments. For example, in 2017, Hamid et al. [13] proposed a triparty authenticated key agreement (AKA) protocol using a fog-computing facility in a healthcare environment. The proposed protocol uses bilinear pairing cryptography and decoy technology to access and store private healthcare data securely. In 2018, Jia et al. [14] proposed a triparty AKA protocol for fog-based healthcare systems. They employed an elliptic curve cryptosystem and bilinear pairing to guarantee the security of the session key. In 2018, Okay et al. [15] described a secure data aggregation protocol for smart grids using fog computing based on the additive privacy scheme proposed by Domingo-Ferrer. Moreover, in 2018, Lyu et al. [16] introduced an efficient and privacy-preserving aggregation scheme using fog-computing architecture to maintain aggregator anonymity. This protocol uses differential privacy and homomorphic encryption to safeguard aggregator obliviousness. In [13,14], computationally expensive cryptographic technology was used for the authentication phase. Bilinear pairing cryptography has high communication costs due to its computational complexity and the additional data required for key generation and data transmission. Moreover, precise data, such as healthcare information, demand high accuracy and reliability, increasing the latency. However, in fog-based VANETs, low latency is vital due to the importance of real-time data transmission and quick decision-making between vehicles. Therefore, the methods in [13,14] are inefficient and unsuitable for fog-based VANETs. In [15,16], the smart-grid environment is based on static data and designed without considering the dynamic network scalability, resulting in a lack of real-time data processing and responsiveness to dynamic situations. However, in fog-based VANETs, where many vehicles move simultaneously, real-time communication and data processing between vehicles and minimizing latency are essential. Therefore, the methods in [15,16] are unsuitable for fog-based VANETs.

Many researchers have studied effective and practical authentication schemes based on fog-based VANETs to address the security and privacy protection demands of vehicle communication. Ma et al. [17] proposed a novel AKA protocol without bilinear pairing to enable secure communication in fog-based VANETs. The protocol offers securely shared session keys, privacy protection, and mutual authentication. Eftekhari et al. [18] suggested a security-enhanced, three-party pairwise shared key agreement protocol for fog-based vehicular communication. They demonstrated that the protocol introduced by Ma et al. [17] does not satisfy several vital security requirements and is vulnerable to security attacks. To address these challenges, they reduced the communication costs compared to the protocol by Ma et al. [17] and improved security by defending against diverse attacks. Kumar et al. [19] introduced an authentication protocol based on fog nodes that adopts a multitrusted authority architecture, using operations based on ECC to achieve low communication and computational costs. They designed a robust and efficient authentication protocol using ECC and symmetric key encryption and decryption systems. Wu et al. [20] designed an authentication key exchange scheme that enhances secure communication in fog-based VANETs with fog nodes as relay nodes. This approach leads to a secure and efficient third-party authentication key exchange scheme. The proposed scheme uses only a few simple operations, including the cryptographic hash function, exclusive OR (XOR), and ECC, considering the restricted computing capabilities of vehicle users and fog nodes. Awais et al. [21] proposed a three-party AKA protocol for fog-based VANETs without depending on bilinear pairing. They used ECC to mitigate security threats in public wireless communication channels. Furthermore, they employed lightweight cryptographic operations for low computational and communication costs. Hedge et al. [8] introduced an efficient and secure authentication scheme using key agreement and management for a cloud–fog-device framework. This scheme applied symmetric trivariate polynomials,

elliptic curve cryptography (ECC), and a fuzzy extractor for authentication. Awais et al. [22] proposed a novel four-party AKA protocol for fog-based VANETs using only lightweight cryptographic techniques and ECC without utilizing bilinear pairing technology. These protocols [8,17–22] proposed an authentication protocol for fog-based VANETs. However, these protocols require high computational and communication costs in order for them to be utilized in a fog-based VANET environment and do not meet several security requirements. Therefore, we proposed a secure and efficient authentication scheme for fog-based VANETs to address these issues.

3. Preliminaries

This section covers the concepts of ECC, the threat model, and the system model illustrated in Figure 1.

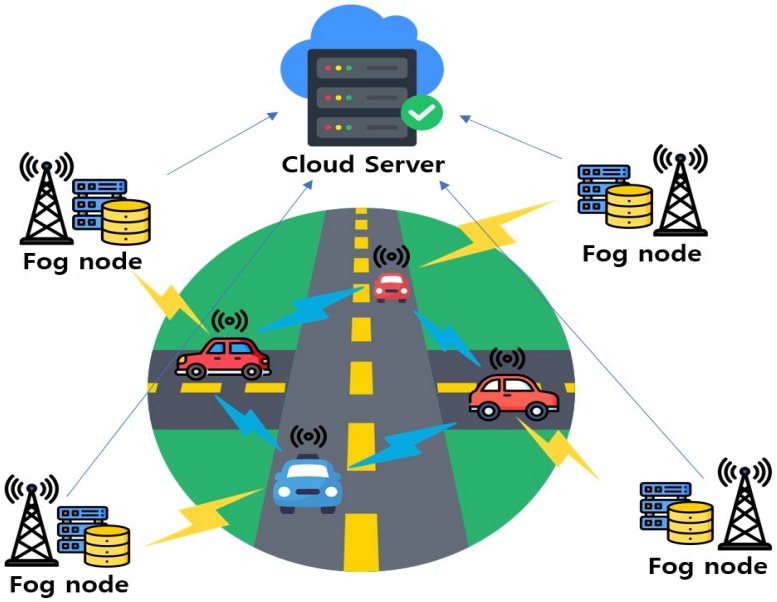

Figure 1. Fog-based VANET architecture.

3.1. Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) [23] is a public key encryption method that applies the mathematical structure of elliptic curves. An elliptic curve, E, is defined as $E_q(a, b)$: $y^2 = x^3 + ax + b(mod p)$, where $a, b \in F_p$, and p and q are large prime, and $4a^3 + 27b^2 \neq 0$. Then, we can select an additive cyclic elliptic curve group, G, with the order q and generator P. The properties of group G are listed below.

- Elliptic Curve Discrete Logarithm Problem: Given two random points, $A, B \in G$, calculating a random value k satisfying $A = k \cdot B$ in polynomial time is infeasible.
- Elliptic Curve Computational Diffie–Hellman (ECCDH) Problem: Given three random points, $A, M, N \in G$, calculating mnA satisfying $M = mA$ and $N = nA$ in polynomial time is infeasible.

3.2. Threat Assumption Model

This paper adopts the Dolev–Yao security model [24–26] and the Canetti and Krawczyk security model [27–29] as threat models for the proposed protocol. The capabilities of an adversary, A, are summarized as follows:

- A can intercept, modify, eavesdrop, and replay messages on public communication channels.

- A and vehicles know the identities of all fog nodes. A may be a legitimate vehicle user or a privileged insider on the cloud server.
- A can obtain the secret values of the smart card through power analysis attacks [30,31].
- A can obtain long- or short-term keys from the network and attempt to compute the session key. The long-term keys are the private keys of the network entities, and the short-term keys are the random values generated during the authentication process [32].

In a real VANET environment, the cloud server and fog nodes are securely connected, making it difficult for attackers to compromise them. However, vehicles can be captured or stolen by attackers, making them vulnerable targets. Therefore, we considered vehicles as insecure entities and assumed a threat model.

3.3. System Model

The system model includes the cloud server (CS), fog node (FN_j), and vehicle user (V_i).

- Cloud server (CS): The CS is a fully trusted entity that initializes the system setup and provides registration services for V_i and FN_j and stores the verification values derived from their identities for authentication.
- Fog Node (FN_j): FN_j is a semi-trusted entity in the protocol that has its own computing capabilities and storage capacity. The fog node mediates the authentication messages transmitted between CS and V_i. Once the authentication phase is complete, FN_j establishes a shared session key with CS and V_i. FN_j has data storage servers and is a wireless communication facility in VANET environments.
- Vehicles (V_i): Each V_i employs its on-board unit to communicate with other vehicles or infrastructure and collect real-time traffic information. In addition, V_i is considered untrustworthy in fog-based VANETs, so the adversary can perform attacks after registering as a legitimate user.

4. Proposed Protocol

This section introduces the proposed protocol comprising five phases: initialization, registration, login and authentication, password update, and user revocation and re-registration. Table 1 lists the notations for the proposed protocol.

Table 1. Notations of the proposed protocol.

Notation	Description
V_i	Vehicle user
FN_j	Fog node
CS	Cloud server
ID_i	Identity of vehicle user
ID_j	Identity of fog node
PID_i	Pseudo identity of vehicle user
RID_i	Secret pseudo identity of vehicle user
PSW_i	Password of vehicle user
N_i	Secret key of vehicle user
N_j	Secret key of fog node
s, x_i, y_j	Secret keys of cloud server
r_i	Set of random numbers
R_i	Set of public keys
SK_{i-j-cs}	Session keys of V_i, FN_j, and CS
$Auth_i, Auth_j$	A secret value needed for authentication
$h(\cdot)$	Cryptographic hash function
$\oplus$	Exclusive OR operation
$\|$	Concatenation operation

4.1. Initialization Phase

CS selects large prime numbers p, q and $a, b \in F_p$. Then, CS selects a secure elliptic curve, $E_q(a, b) : y^2 = x^3 + ax + b$ $(4a^3 + 27b^2 \neq 0)$, in a finite field, F_p, and $t = log_{2p}$ represents the security metrics. Moreover, G denotes a cyclic group with order q with a base point P. Then, CS randomly selects an integer, $s \in Z_q^*$, as a secret key and computes $P_{pub} = sP$. The public system parameters are released as (G, P, P_{pub}), whereas the value of s remains confidential. Then, CS selects secure one-way hash functions $h(.) : \{0,1\}^* \rightarrow \{0,1\}^{tn}$, generating a 256-bit output and a fuzzy verifier, $2^4 \leq s_0 \leq 2^8$. Finally, CS publishes system parameters $\{P_{pub}, E_q, P, h(.)\}$ and keeps s secret.

4.2. Vehicle Registration Phase

Each vehicle transmits a registration request to a fully trusted cloud server in this phase and receives a smart card.

Step 1: V_i inputs a user identity (ID_i) and password, PSW_i, and chooses an integer, $2^4 \leq s_0 \leq 2^8$. V_i computes $RPSW_i = h(ID_i \parallel PSW_i \parallel s_0)$ and $PID_i = h(ID_i \parallel RPSW_i \parallel s_0)$ and sends (ID_i) to CS in a secure manner.

Step 2: After responding to the request of V_i, CS randomly selects $x_i \in Z_p^*$ and calculates $N_i = h(PID_i \parallel s \parallel x_i)$. Then, CS stores N_i on a smart card and sends it to V_i through a secure channel. CS also stores the pair (PID_i, x_i) in a database.

Step 3: Then, V_i computes $M_i = h((h(ID_i) \oplus PSW_i) mods_0)$ and $N_i^* = h(RPSW_i \parallel M_i) \oplus N_i$. V_i stores (N_i^*, M_i, s_0) on a smart card and deletes N_i.

Figure 2 presents the vehicle registration phase.

Vehicle user (V_i)		Cloud server (CS)
Inputs ID_i and PSW_i		
Chooses s_0		
Computes $RPSW_i = h(ID_i \parallel PSW_i \parallel s_0)$		
$PID_i = h(ID_i \parallel RPSW_i \parallel s_0)$		
	$\xrightarrow[\text{Secure channel}]{PID_i}$	
		Selects $x_i \in Z_p^*$
		Calculates $N_i = h(PID_i \parallel s \parallel x_i)$
		Store N_i in smart card
		Stores the pair (PID_i, x_i)
	$\xleftarrow[\text{Secure channel}]{Smartcard}$	
Computes $M_i = h((h(ID_i \oplus PSW_i) mods_0)$		
$N_i^* = h(RPSW_i \parallel M_i) \oplus N_i$		
Stores (N_i^*, M_i, s_0) in smart card		
Deletes N_i in smart card		

Figure 2. Proposed vehicle registration phase.

4.3. Fog Node Registration Phase

FN_j is registered with CS before deployment. To achieve this, FN_j transmits its identity ID_j to CS, which randomly selects y_j from the set of integers, where $y_j \leq Z_p^*$. Afterward, CS computes $N_j = h(ID_j \parallel s \parallel y_j)$ and securely transmits N_j to FN_j and stores the pair (ID_j, y_j) in its database. Figure 3 presents the fog node registration phase.

Fog node (FN_j)		Cloud server (CS)
Selects ID_j		
	$\xrightarrow[\text{Secure channel}]{ID_j}$	
		Selects $y_j \in Z_p^*$ Calculates $N_j = h(ID_j \parallel s \parallel y_j)$ Stores the pair (ID_j, y_j)
	$\xleftarrow[\text{Secure channel}]{N_j}$	

Figure 3. Proposed fog node registration phase.

4.4. Login and Authentication Phase

In this phase, CS, FN_j, and V_i authenticate each other using their secret values and agree on a shared session key for secure communication. Figure 4 presents their interactions, and the details are provided below.

Step 1: Insert ID_i and PSW_i and compute $M_i' = h((h(ID_i) \oplus PSW_i) mods_0)$. After computing, check whether $M_i' = M_i$; if not true, V_i will terminate the session and notify the user of the login failure. Then, V_i request the user to retry the login process. If true, V_i computes $RPSW_i' = h(ID_i \parallel PSW_i \parallel s_0)$ and $N_i = N_i^* \oplus h(RPSW_i' \parallel M_i')$. Next, V_i selects a random number, $r_1 \leq Z_q^*$, and calculates $R_1 = r_1 P, \bar{R}_1 = r_1 P_{pub}, RID_i = PID_i \oplus h(\bar{R}_1)$, and $Q_i = h(\bar{R}_1 \parallel N_i \parallel ID_i \parallel ID_j)$, and V_i transmits $(D_1 = R_1, RID_i, Q_i, ID_j)$ to FN_j.

Step 2: FN_j verifies the freshness of the random number in D_1 from V_i and selects a random number, $r_2 \in Z_q^*$, to calculate $R_2 = r_2 P$ and $L_j = h(N_j \parallel ID_j \parallel Q_i)$. FN_j then transmits $(D_2 = R_1, RID_i, R_2, r_2 R_1, L_j, ID_j)$ to CS.

Step 3: After receiving the authentication request from FN_j, CS verifies the freshness of the random number in D_2 from FN_j and computes $\bar{R}_1' = sR_1$ and $PID_i' = RID_i \oplus h(\bar{R}_1')$. CS checks its database for items that correspond to (PID_i', x_i) and (ID_j', y_j). If CS does not find such items, it rejects the request and terminates the session. If CS finds the items, CS continues with additional computations as follows: - $N_i' = h(ID_i' \parallel s \parallel x_i)$ - $N_j' = h(ID_j \parallel s \parallel y_j)$ - $Q_i' = h(\bar{R}_1' \parallel N_i' \parallel ID_i' \parallel ID_j)$ - $L_j' = h(N_j \parallel ID_j \parallel Q_i')$. CS checks whether $L_j' = L_j$ is true, and if conditions are not true, CS will terminate the current session and request FN_j and V_i to retry the authentication process. Otherwise, CS randomly chooses $r_3 \in Z_q^*$, calculates $R_3 = r_3 P$, and computes the following: - $R_{i-j-cs} = r_3 \cdot (r_2 R_1)$ - $K_{ij} = h(N_i \parallel R_{i-j-cs}) \oplus h(N_j \parallel R_{i-j-cs})$ - $SK_{i-j-cs} = h(h(N_i \parallel R_{i-j-cs}) \parallel h(N_j \parallel R_i i-j-cs) \parallel R_{i-j-cs})$ - $Auth_j = h(h(N_i \parallel R_{i-j-cs}) \parallel R_{i-j-cs} \parallel SK_{i-j-cs})$. Then, CS transmits $(D_3 = K_{ij}, R_3, r_3 R_1, Auth_j)$ to FN_j.

Step 4: After receiving the response from CS, FN_j verifies the freshness of the random number in D_3 from CS and computes $R_{i-j-cs} = r_2 \cdot (r_3 R_1), h(N_i \parallel R_{i-j-cs}) = K_{ij} \oplus h(N_j \parallel R_{i-j-cs}), SK_{i-j-cs} = h(h(N_i \parallel R_{i-j-cs}) \parallel h(N_j \parallel R_{i-j-cs}) \parallel R_{i-j-cs})$, and $Auth_j' = h(h(N_i \parallel R_{i-j-cs}) \parallel R_{i-j-cs} \parallel SK_{i-j-cs})$. Then, FN_j verifies whether or not $Auth_j' = Auth_j$ is true, and if conditions are not true, FN_j will terminate the current session and notify CS of authentication failure. If true, FN_j computes $Auth_i = h(Auth_j \parallel h(N_j \parallel R_{i-j-cs}))$ and transmits $(D_4 = K_{ij}, r_2 R_3, Auth_i)$ to V_i.

Step 5: After receiving the response of FN_j, V_i verifies the freshness of the random number in D_4 from FN_j and computes $R_{i-j-cs} = r_1 \cdot (r_2 R_3), h(N_j \parallel R_{i-j-cs}) = K_{ij} \oplus h(N_i \parallel R_{i-j-cs}), SK_{i-j-cs} = h(h(N_i \parallel R_{i-j-cs}) \parallel h(N_j \parallel R_{i-j-cs}) \parallel R_{i-j-cs}), Auth_j' =$

$h(h(N_i \parallel R_{i-j-cs}) \parallel R_{i-j-cs} \parallel SK_{i-j-cs})$, and $Auth_i' = h(Auth_j' \parallel h(N_j \parallel R_{i-j-cs}))$. Then, V_i verifies whether or not $Auth_i' = Auth_i$; if conditions are not true, V_i will terminate the current session.

After the authentication phase is fully completed, a secure shared session key is established between V_i, FN_j, and CS.

Vehicle user (V_i)	Fog node (FN_j)	Cloud server (CS)
% Login Inserts ID_i, PSW_i Computes $M_i' = h((h(ID_i) \oplus PSW_i) mod s_0)$ Checks $M_i' = M_i$ If not true, rejects login try; % Generate authentication message Otherwise, computes $RPSW_i' = h(ID_i \parallel PSW_i \parallel s_0)$ $N_i = N_i^* \oplus h(RPSW_i' \parallel M_i')$ $r_1 \in Z_q^*, R_1 = r_1 P, \bar{R}_1 = r_1 P_{pub}$ $RID_i = PID_i \oplus h(\bar{R}_1)$ $Q_i = h(\bar{R}_1 \parallel N_i \parallel ID_i \parallel ID_j)$ $(D_1{=}R_1, RID_i, Q_i, ID_j) \longrightarrow$		
	% Relay authentication message $r_2 \in Z_q^*, R_2 = r_2 P$ $L_j = h(N_j \parallel ID_j \parallel Q_i)$ $(D_2{=}R_1, RID_i, R_2, r_2 R_1, L_j, ID_j) \longrightarrow$	
		% Verify authentication message $\bar{R}_1' = s\bar{R}_1$ Computes $PID_i' = RID_i \oplus h(\bar{R}_1')$ Search the entire $(PID_i', x_i), (ID_j, y_j)$ in database If not found, rejects the request and terminates the session Otherwise, computes $N_i' = h(ID_i' \parallel s \parallel x_i)$ $N_j = h(ID_j \parallel s \parallel y_i)$ $Q_i' = h(\bar{R}_1' \parallel N_i' \parallel ID_i' \parallel ID_j)$ $L_j' = h(N_j \parallel ID_j \parallel Q_i')$ Checks $L_j' = L_j$ If not true, terminates the session; % Generate response message Otherwise, chooses $r_3 \in Z_q^*, R_3 = r_3 P$ Computes $R_{i-j-cs} = r_3 \cdot (r_2 R_1)$ $K_{ij} = h(N_i \parallel R_{i-j-cs}) \oplus h(N_j \parallel R_{i-j-cs})$ $SK_{i-j-cs} = h(h(N_i \parallel R_{i-j-cs}) \parallel h(N_j \parallel R_{i-j-cs}) \parallel R_{i-j-cs})$ $Auth_j = h(h(N_i \parallel R_{i-j-cs}) \parallel R_{i-j-cs} \parallel SK_{i-j-cs})$ $\longleftarrow (D_3{=}K_{ij}, R_3, r_3 R_1, Auth_j)$
	% Relay response message Computes $R_{i-j-cs} = r_2 \cdot (r_3 R_1)$ $h(N_i \parallel R_{i-j-cs}) = K_{ij} \oplus h(N_j \parallel R_{i-j-cs})$ $SK_{i-j-cs} = h(h(N_i \parallel R_{i-j-cs}) \parallel h(N_j \parallel R_{i-j-cs}) \parallel R_{i-j-cs})$ $Auth_j' = h(h(N_i \parallel R_{i-j-cs}) \parallel R_{i-j-cs} \parallel SK_{i-j-cs})$ Checks $Auth_j' = Auth_j$ If not true, terminates the session; Otherwise, computes $Auth_i = h(Auth_j \parallel h(N_j \parallel R_{i-j-cs}))$ $\longleftarrow (D_4{=}K_{ij}, r_2 R_3, Auth_i)$	
% Verify response message Computes $R_{i-j-cs} = r_1 \cdot (r_2 R_3)$ $h(N_j \parallel R_{i-j-cs}) = K_{ij} \oplus h(N_i \parallel R_{i-j-cs})$ $SK_{i-j-cs} = h(h(N_i \parallel R_{i-j-cs}) \parallel h(N_j \parallel R_{i-j-cs}) \parallel R_{i-j-cs})$ $Auth_j' = h(h(N_i \parallel R_{i-j-cs}) \parallel R_{i-j-cs} \parallel SK_{i-j-cs})$ $Auth_i' = h(Auth_j' \parallel h(N_j \parallel R_{i-j-cs}))$ Checks $Auth_i' = Auth_i$ If not true, terminates the session;		

Figure 4. Proposed authentication phase.

4.5. Password Update Phase

Vehicle users can update their passwords as often as desired, as follows:

Step 1: After entering ID_i and PSW_i in the smart card, V_i sends a password change request.

Step 2: The smart card computes $RPSW_i = h(ID_i \parallel PSW_i \parallel s_0)$ and $M_i = h(h(ID_i) \oplus PSW_i) mods_0)$ and then checks whether or not $M_i' = M_i$ matches the stored M_i to verify the authenticity of M_i'. If the verification is confirmed, the smart card inputs ID_i and PSW_i.

Step 3: First, the new password, PSW_i^{new}, must be entered. Then, the smart card generates $2^4 \leq s_0^{new} \leq 2^8$ and calculates $RPSW_i^{new} = h(ID_i \parallel PSW_i^{new} \parallel s_o^{new})$ and $M_i^{new} = h(h(ID_i) \oplus PSW_i^{new}) mods_0^{new})$. Finally, the smart card replaces (N_i^*, M_i, s_0) with $(N_i^{new}, M_i^{new}, s_0^{new})$.

4.6. User Revocation and Re-Registration

If V_i is compromised, CS deletes (ID_i, x_i) from its database. Then, login attempts with the previous smart card are rejected.

CS allows V_i to re-register through registration. V_i re-registers using the same identity and an updated password. Afterward, CS assigns a new random number, x_i^{new}, and stores it in the database with V_i's PID_i.

4.7. Fog Node Revocation

If FN_j is compromised, CS deletes (ID_j, y_j) from its database. Access requests from FN_j are denied afterward because the random number y_j is needed to verify authentication requests.

5. Security Analysis

This section analyzes the proposed protocol using formal and informal methods.

5.1. Formal Analysis

5.1.1. BAN Logic

This section presents the Burrows–Abadi–Nikoogadam (BAN) logic [33] of the proposed protocol. The BAN logic method is a formal approach used for analyzing and verifying the correctness of authentication protocols. Table 2 presents the notation and definitions, and the BAN logic rules are provided below.

Table 2. Burrows –Abadi–Nikoogadam (BAN) logic notation.

Notation	Description
θ_1, θ_2	Principals
σ_1, σ_2	Statements
$\theta_1 \mid \equiv \sigma_1$	θ_1 **believes** σ_1
$\theta_1 \mid \sim \sigma_1$	θ_1 once **said** σ_1
$\theta_1 \Rightarrow \sigma_1$	θ_1 **controls** σ_1
$\theta_1 \lhd \sigma_1$	θ_1 **receives** σ_1
$\#\sigma_1$	σ_1 is **fresh**
$(\sigma_1)_K$	σ_1 is **encrypted** by K
$\theta_1 \overset{K}{\leftrightarrow} \theta_2$	θ_1 and θ_2 have **shared** key K
SK	Session key

5.1.2. Burrows–Abadi–Nikoogadam (BAN) Logic Rules

1. Message meaning rule (MMR):

$$\frac{\theta_1 \mid \equiv \theta_1 \overset{K}{\leftrightarrow} \theta_2, \quad \theta_1 \lhd (\sigma_1)_K}{\theta_1 \mid \equiv \theta_2 \mid \sim \sigma_1}$$

2. Nonce verification rule (NVR):

$$\frac{\theta_1 \mid \equiv \#(\sigma_1), \quad \theta_1 \mid \equiv \theta_2 \mid \sim \sigma_1}{\theta_1 \mid \equiv \theta_2 \mid \equiv \sigma_1}$$

3. Jurisdiction rule (JR):

$$\frac{\theta_1 \mid \equiv \theta_2 \mid \Longrightarrow \sigma_1, \quad \theta_1 \mid \equiv \theta_2 \mid \equiv \sigma_1}{\theta_1 \mid \equiv \sigma_1}$$

4. Belief rule (BR):

$$\frac{\theta_1 \mid \equiv (\sigma_1, \sigma_2)}{\theta_1 \mid \equiv \sigma_1}$$

5. Freshness rule (FR):

$$\frac{\theta_1 \mid \equiv \#(\sigma_1)}{\theta_1 \mid \equiv \#(\sigma_1, \sigma_2)}$$

5.1.3. Goals

The goals are to demonstrate that the vehicle user, V_i, fog node, FN_j, and cloud server, CS, all agree on the same session key, SK.

G 1: $V_i \mid \equiv V_i \overset{SK}{\longleftrightarrow} FN_j$

G 2: $V_i \mid \equiv FN_j \mid \equiv V_i \overset{SK}{\longleftrightarrow} FN_j$

G 3: $FN_j \mid \equiv V_i \overset{SK}{\longleftrightarrow} FN_j$

G 4: $FN_j \mid \equiv V_i \mid \equiv V_i \overset{SK}{\longleftrightarrow} FN_j$

G 5: $FN_j \mid \equiv CS \overset{SK}{\longleftrightarrow} FN_j$

G 6: $FN_j \mid \equiv CS \mid \equiv CS \overset{SK}{\longleftrightarrow} FN_j$

G 7: $CS \mid \equiv CS \overset{SK}{\longleftrightarrow} FN_j$

G 8: $CS \mid \equiv FN_j \mid \equiv CS \overset{SK}{\longleftrightarrow} FN_j$

5.1.4. Idealized Forms

The following idealized forms of each message are transmitted during the authentication phase:

D_1 : $V_i \rightarrow FN_j : (R_1, Q_i)_{N_i}$

D_2 : $FN_j \rightarrow CS : (R_1, R_2, r_2 R_1, L_j)_{N_j}$

D_3 : $CS \rightarrow FN_j : (R_3, r_3 R_1, h(N_i \parallel R_{i-j-cs}))_{h(N_j \| R_{i-j-cs})}$

D_4 : $FN_j \rightarrow V_i : (r_2 R_3, h(N_j \parallel R_{i-j-cs}))_{h(N_i \| R_{i-j-cs})}$

5.1.5. Assumptions

The assumptions of the proposed protocol are provided below.

A_1: $V_i| \equiv \#(r_2 R_3)$

A_2: $FN_j| \equiv \#(R_1)$

A_3: $FN_j| \equiv \#(R_3)$

A_4: $CS| \equiv \#(R_2)$

A_5: $V_i| \equiv FN_j \Rightarrow (V_i \overset{SK}{\longleftrightarrow} FN_j)$

A_6: $FN_j| \equiv CS \Rightarrow (CS \overset{SK}{\longleftrightarrow} FN_j)$

A_7: $CS| \equiv FN_j \Rightarrow (CS \overset{SK}{\longleftrightarrow} FN_j)$

A_8: $FN_j| \equiv V_i \Rightarrow (V_i \overset{SK}{\longleftrightarrow} FN_j)$

A_9: $V_i| \equiv V_i \overset{h(N_i \| R_{i-j-cs})}{\longleftrightarrow} FN_j$

A_{10}: $FN_j| \equiv CS \overset{h(N_j)}{\longleftrightarrow} FN_j$

A_{11}: $CS| \equiv CS \overset{h(N_j)}{\longleftrightarrow} FN_j$

A_{12}: $FN_j| \equiv V_i \overset{h(N_i \| R_{i-j-cs})}{\longleftrightarrow} FN_j$

A_{13}: $V_i| \equiv V_i \overset{h(N_i)}{\longleftrightarrow} CS$

A_{14}: $CS| \equiv V_i \overset{h(N_i)}{\longleftrightarrow} CS$

5.1.6. Burrows–Abadi–Nikoogadam (BAN) Logic Proof

The BAN logic proof is based on the following assumptions and idealized forms:

S 1: FN_j receives D_1.

$$S_1\text{: } FN_j \lhd (R_1, Q_i)_{N_i}$$

S 2: CS receives D_2.

$$S_2\text{: } CS \lhd (R_1, R_2, r_2 R_1, L_j)_{N_j}$$

S 3: Applying S_2 and A_{11} to the MMR yields S_3.

$$S_3 : CS| \equiv FN_j| \sim (R_1, R_2, r_2 R_1, L_j)$$

S 4: Applying S_3 and A_4 to the FR yields S_8.

$$S_4 : CS| \equiv \#(R_1, R_2, r_2 R_1, L_j)$$

S 5: Applying S_3 and S_4 to the NVR yields S_5.

$$S_5 : CS| \equiv FN_j| \equiv (R_1, R_2, r_2 R_1, L_j)$$

S 6: We can obtain S_6 by applying S_5 to the BR.

$$S_6 : CS| \equiv FN_j| \equiv (r_2 R_1)$$

S 7: FN_j receives D_3.

$$S_7\text{: } FN_j \lhd (R_3, r_3R_1, h(N_i \parallel R_{i-j-cs}))_{h(N_j \parallel R_{i-j-cs})}$$

S 8: Applying S_7 and A_{10} to the MMR yields S_8.

$$S_8 : FN_j| \equiv CS| \sim (R_3, r_3R_1, h(N_i \parallel R_{i-j-cs}))$$

S 9: Applying S_8 and A_3 to the FR yields S_9.

$$S_9 : FN_j| \equiv \#(R_3, r_3R_1, h(N_i \parallel R_{i-j-cs}))$$

S 10: Applying S_8 and S_9 to the NVR yields S_{10}.

$$S_{10} : FN_j| \equiv CS| \equiv (R_3, r_3R_1, h(N_i \parallel R_{i-j-cs}))$$

S 11: We can obtain S_{11} by applying S_{10} to the BR.

$$S_{11} : FN_j| \equiv CS| \equiv (r_3R_1)$$

S 12: V_i receives D_4.

$$S_{12}\text{: } V_i \lhd (r_2R_3, h(N_j \parallel R_{i-j-cs}))_{h(N_i \parallel R_{i-j-cs})}$$

S 13: Applying S_{12} and A_9 to the MMR yields S_{13}.

$$S_{13} : V_i| \equiv FN_j| \sim (r_2R_3, h(N_j \parallel R_{i-j-cs}))$$

S 14: Applying S_{13} and A_1 to the FR yields S_{14}.

$$S_{14} : V_i| \equiv \#(r_2R_3, h(N_j \parallel R_{i-j-cs}))$$

S 15: Applying S_{13} and S_{14} to the NVR yields S_{15}.

$$S_{15} : V_i| \equiv FN_j| \equiv (r_2R_3, h(N_j \parallel R_{i-j-cs}))$$

S 16: We can obtain S_{16} by applying S_{15} to the BR.

$$S_{16} : V_i| \equiv FN_j| \equiv (r_2R_3)$$

S 17: From S_6, S_{11}, and S_{16}, V_i, FN_j, and CS can compute the session key $SK_{i-j-cs} = h(h(N_i \parallel R_{i-j-cs}) \parallel h(N_j \parallel R_{i-j-cs}) \parallel R_{i-j-cs})$.

$$S_{17}\text{: } V_i| \equiv FN_j| \equiv V_i \overset{SK}{\longleftrightarrow} FN_j \quad \textbf{(Goal 2)}$$
$$S_{18}\text{: } FN_j| \equiv V_i| \equiv V_i \overset{SK}{\longleftrightarrow} FN_j \quad \textbf{(Goal 4)}$$
$$S_{19}\text{: } FN_j| \equiv CS| \equiv CS \overset{SK}{\longleftrightarrow} FN_j \quad \textbf{(Goal 6)}$$
$$S_{20}\text{: } CS| \equiv FN_j| \equiv CS \overset{SK}{\longleftrightarrow} FN_j \quad \textbf{(Goal 8)}$$

S 18: The JR can be applied to S_{21}, S_{22}, S_{23}, and S_{24} using A_5, A_8, A_6, and A_7, respectively.

$$S_{21}\text{: } V_i| \equiv V_i \overset{SK}{\longleftrightarrow} FN_j \quad \textbf{(Goal 1)}$$
$$S_{22}\text{: } FN_j| \equiv V_i \overset{SK}{\longleftrightarrow} FN_j \quad \textbf{(Goal 3)}$$
$$S_{23}\text{: } FN_j| \equiv CS \overset{SK}{\longleftrightarrow} FN_j \quad \textbf{(Goal 5)}$$
$$S_{24}\text{: } CS| \equiv CS \overset{SK}{\longleftrightarrow} FN_j \quad \textbf{(Goal 7)}$$

Finally, the vehicle user, fog node, and cloud server mutually authenticate each other.

5.1.7. Real-or-Random Model

The real-or-random (RoR) model [34] is a formal security analysis method that proves the semantic security of the session key in the authentication protocol. In the proposed protocol, the participants are the vehicle user, fog node, and cloud server: $Ta_{V_i}^{k_1}$, $Ta_{FN_i}^{k_2}$, and $Ta_{CS}^{k_3}$, respectively. In the RoR model, adversary A can intercept, eavesdrop, replay, and modify all insecure channel messages to determine the session key, SK. A can perform the queries $Execute(Ta_{V_i}^{k_1}, Ta_{FN_i}^{k_2}, Ta_{CS}^{k_3})$, $CorruptSC(Ta_{V_i}^{k_1})$, $Send(Ta_x^{k_n}, Msg)$, and $Test(Ta_x^{k_n})$. Table 3 presents the queries performed by A.

Table 3. Queries in the real-or-random (RoR) model.

Query	Description
$Execute(Ta_{V_i}^{k_1}, Ta_{FN_i}^{k_2}, Ta_{CS}^{k_3})$	A can eavesdrop messages transmitted via public channels between $Ta_{V_i}^{k_1}$, $Ta_{FN_i}^{k_2}$, and $Ta_{CS}^{k_3}$. A can perform passive attacks with these messages.
$CorruptSC(Ta_{V_i}^{k_1})$	A can obtain secret values stored in the stolen smart card of $Ta_{V_i}^{k_1}$ by performing this query.
$Send(Ta_x^{k_n}, Msg)$	By performing this query, A can send a message, Msg, to a participant, $Ta_x^{k_n}$. Furthermore, A can obtain a response message from a participant, $Ta_x^{k_n}$.
$Test(Ta_x^{k_n})$	In the last game, A performs this query. When this query is performed, an unbiased coin, c, is tossed. The head represents 1 and the tail represents 0. If $c = 1$, then $Ta_x^{k_n}$ returns the session key, SK; If $c = 0$, then $Ta_x^{k_n}$ returns a random number. In other cases, $Ta_x^{k_n}$ returns $NULL$. If A correctly guesses that the returned value is the session key, SK, A wins the game.

Theorem 1: We define q_{ha}, $|Hash|$, q_{send}, and l as the number of hash queries performed by A, the range space of the hash function, the number of send queries performed by A, and the length of the identity V_i, respectively. Furthermore, the breaking possibility of the ECCDH problem is $Advp_M^{ECC}(A)$, and the Zipf parameters are C' and s'. When $Advp(A)$ is the probability that A breaks the session key in polynomial time, we prove the following equation:

$$Advp(A) \leq \frac{q_{ha}^2}{|Hash|} + 2Advp_M^{ECC}(A) + 2max\{C' q_{send}^{s'}, \frac{q_{send}}{2^l}\} \quad (1)$$

Proof. A plays five games, $GM_n(n = 0, 1, 2, 3, 4)$, based on the RoR model. $AVTG^{WIGM_n}(A)$ represents the advantage of A to break the session key after playing the game GM_n.

- GM_0: In the first game, A selects a random bit, r. A does not know any information required to calculate the session key, SK, and has no queries to perform. Thus, we derive the following equation:

$$Advp(A) = |2AVTG_{WIGM_0}(A) - 1|. \quad (2)$$

- GM_1: A performs the $Execute$ query to conduct an eavesdropping attack. From that query, A obtains all public channel messages $(D_1 = R_1, RID_i, Q_i, ID_j)$, $(D_2 = R_1, RID_i, R_2, r_2R_1, L_j, ID_j)$, $(D_3 = K_{ij}, R_3, r_3R_1, Auth_j)$, and $(D_4 = K_{ij}, r_2R_3, Auth_i)$. Afterward, A performs a $Test$ query to calculate the session key, SK. However, A cannot calculate the session key, SK, because it is masked by long-term keys N_i and N_j and the short-term key R_{i-j-cs}. Thus, we obtain the following equation:

$$AVTG_{WIGM_1}(A) = AVTG_{WIGM_0}(A). \tag{3}$$

- GM_2: In this game, A performs the *Send* and *Hash* queries to calculate the session key, SK. To obtain the values needed for calculating the session key, SK, A must determine the hash collision using messages from the public channel. Thus, we obtain the following equation due to the birthday paradox [35]:

$$|AVTG_{WIGM_2}(A) - AVTG_{WIGM_1}(A)| \leq \frac{q_{ha}^2}{2|Hash|}. \tag{4}$$

- GM_3: In this game, A tries to compute SK with the messages $(D_1 = R_1, RID_i, Q_i, ID_j)$, $(D_2 = R_1, RID_i, R_2, r_2R_1, L_j, ID_j)$, $(D_3 = K_{ij}, R_3, r_3R_1, Auth_j)$, and $(D_4 = K_{ij}, r_2R_3, Auth_i)$. However, the session key, SK, consists of $R_{i-j-cs} = r_3 \cdot r_2 \cdot r_1 \cdot P$, derived from the ECCDH problem. Therefore, we obtain the following inequality:

$$|AVTG_{WIGM_3}(A) - AVTG_{WIGM_2}(A)| \leq Advp_M^{ECC}(A). \tag{5}$$

- GM_4: In the last game, A performs the *CorruptSC* query and extracts (N_i^*, M_i, s_0) from SC. However, A cannot compute the session key, SK, because the smart card values are masked with a hash function using ID_i and PSW_i. Thus, we obtain the following inequality using the Zipf law [36]:

$$|AVTG_{WIGM_4}(A) - AVTG_{WIGM_3}(A)| \leq max\{C' q_{send}^{s'}, \frac{q_{send}}{2^l}\}. \tag{6}$$

At the end of all games, A must guess whether or not r is correct from the *Test* query. Therefore, we obtain the following equation:

$$AVTG_{WIGM_4}(A) = \frac{1}{2}. \tag{7}$$

We derive the following equation from Equation (2) and (3):

$$\begin{aligned} \frac{1}{2}Advp(A) &= |AVTG_{WIGM_0}(A) - \frac{1}{2}| \\ &= |AVTG_{WIGM_1}(A) - \frac{1}{2}|. \end{aligned} \tag{8}$$

We also compute the following equation from Equations (7) and (8):

$$\frac{1}{2}Advp(A) = |AVTG_{WIGM_1}(A) - AVTG_{WIGM_4}(A)|. \tag{9}$$

We apply the triangular inequality to the Equation (9).

$$\begin{aligned} \frac{1}{2}Advp(A) &= |AVTG_{WIGM_1}(A) - AVTG_{WIGM_4}(A)| \\ &\leq |AVTG_{WIGM_1}(A) - AVTG_{WIGM_3}(A)| \\ &+ |AVTG_{WIGM_3}(A) - AVTG_{WIGM_4}(A)| \\ &\leq |AVTG_{WIGM_1}(A) - AVTG_{WIGM_2}(A)| \\ &+ |AVTG_{WIGM_2}(A) - AVTG_{WIGM_3}(A)| \\ &+ |AVTG_{WIGM_3}(A) - AVTG_{WIGM_4}(A)| \\ &\leq \frac{q_{ha}^2}{2|Hash|} + Advp_M^{ECC}(A) + max\{C' q_{send}^{s'}, \frac{q_{send}}{2^l}\}. \end{aligned} \tag{10}$$

By multiplying Equation (10) by two, we obtain the following result:

$$Advp(A) \leq \frac{q_{ha}^2}{|Hash|} + 2Advp_M^{ECC}(A) + 2max\{C'q_{send}^{s'}, \frac{q_{send}}{2^l}\}. \quad (11)$$

Finally, we prove the semantic security of our proposed protocol using the RoR model. □

5.1.8. AVISPA Simulation

In this section, we presents formal security verification of our proposed protocol using AVISPA [37]. AVISPA has been widely used to verify the security of authentication protocols, primarily to assess their resilience against man-in-the-middle and replay attacks. Moreover, AVISPA is a formal analysis tool that implements an authentication protocol using the High-Level Protocol Specification Language (HLPSL). In addition, AVISPA uses four back-end models: "on-the-fly model-checker (OFMC)", "constraint logic-based attack searcher (CL-AtSe)", "SAT-based model-checker (SATMC)", and "tree automata based on automatic approximations for the analysis of security protocols (TA4SP)". The back-end models evaluate the security features of an authentication protocol and generate the output format as a result. Since XOR operation is used in our proposed protocol, we only use OFMC and CL-AtSe back-end models. Figure 5 depicts the simulation results, and the proposed protocol is considered safe. Therefore, we can demonstrate that the proposed protocol resists replay and man-in-the-middle (MITM) attacks.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/fogauthenti.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.00s
  visitedNodes: 8 nodes
  depth: 3 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/fogauthenti.if

GOAL
  as_specified

BACKEND
  OFMC

STATISTICS

  Analysed: 3 states
  Reachable: 0 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

Figure 5. Simulation results under OFMC and CL-AtSe.

5.2. Informal Analysis

This section demonstrates that the proposed protocol satisfies the security properties detailed below.

5.2.1. Anonymity and Untraceability

In the proposed protocol, the vehicle user performs the authentication step using the pseudo identity, PID_i, and the temporary identity, RID_i, ensuring the anonymity of the vehicle user. Furthermore, PID_i and RID_i change dynamically with each session due to timestamps and random values; hence, the attacker cannot track the vehicle user. Therefore, the proposed protocol guarantees the anonymity and untraceability of the vehicle user.

5.2.2. Perfect Forward Secrecy

If the long-term keys s, x_i, and y_j of the cloud server are exposed to an attacker, the attacker can compute the long-term keys of the vehicle and the fog node, N_i and N_j. However, the attacker cannot recognize both the long- and short-term keys of the network simultaneously (Section 3.2) because, even if the attacker knows the values of N_i and N_j, the attacker cannot determine the value of R_{i-j-cs}, consisting of the random values r_1, r_2, and r_3. Therefore, the attacker cannot calculate the session key, comprising the long-term keys N_i and N_j and short-term key R_{i-j-cs}. Therefore, the proposed protocol can safeguard perfect forward secrecy.

5.2.3. Stolen-Verifier Attack

If the cloud server database is leaked to the attacker, the attacker can obtain the pseudo identity and the random value x_i of the vehicle user and the identity and random value y_j of the fog node. The attacker may endeavor to calculate the session key using these values. However, without knowing the short-term keys r_1, r_2, and r_3, the attacker cannot compute the session key. Furthermore, the pseudo identity of the vehicle is masked with its identity, password, and s_0, and the attacker cannot derive sensitive information from the identity and the random values x_i and y_j of the fog node.

5.2.4. Stolen Smart Card Attack

The attacker can steal the vehicle's smart card to obtain the stored data (N_i^*, M_i, s_0). Based on these parameters, the attacker may attempt to impersonate the vehicle user and calculate the session key. However, all parameters are masked with the ID, password, and s_0 value of V_i. The attacker must guess the ID and password simultaneously, which is computationally infeasible. Therefore, the proposed protocol is resistant to stolen smart card attacks.

5.2.5. Session Key Disclosure Attack

The attacker may attempt to determine the session key using messages from the public channel and the obtained values. However, to compute the session key, the attacker must guess the values of N_i, N_j, and R_{i-j-cs}, which are masked in a hash function using s, xi, yj, and random values. The attacker cannot obtain these values; thus, the proposed protocol resists session key disclosure attacks.

5.2.6. Replay and Man-in-the-Middle Attacks

To attempt a replay attack, the attacker may intercept public channel messages D_1, D_2, D_3, and D_4. However, these messages contain timestamps T_1, T_2, T_3, and T_4 and verification parameters Q_i, L_j, $Auth_i$, and $Auth_j$, and each entity checks the freshness of the messages. Therefore, the network participants can confirm the secret parameter values. Thus, the proposed protocol resists replay and MITM attacks.

5.2.7. Ephemeral Secret Leakage Attack

If the temporary secret random values r_1, r_2, and r_3 are leaked, the attacker may attempt to calculate the value of R_{i-j-cs}. However, the attacker cannot know the long- and short-term keys of the network simultaneously (as seen in the threat model assumptions); thus, even if the attacker knows the random values r_1, r_2, and r_3, they cannot determine the values of N_i and N_j. Therefore, the attacker cannot compute the session key, comprising the long-term keys N_i and N_j and short-term key R_{i-j-cs}. Therefore, the proposed protocol resists ephemeral leakage attacks.

5.2.8. Privileged Insider Attack

If the attacker acquires all values used during the registration process, they may attempt to guess the ID and password of the vehicle user. However, the values used during the vehicle registration phase are masked in a hash function, including the ID, password, and s_0 values, making it impossible to guess them simultaneously. Therefore, the proposed protocol is resilient to privileged insider attacks.

6. Performance Analysis

In this section, the computational and communication cost of our protocol is compared to that of existing related protocols [18–21].

6.1. Computational Cost Analysis

This subsection compares the computational cost of our proposed protocol with [8,17–22]. We denoted the consumption time of the ECC scalar multiplication, hash operation, symmetric cryptography operation, and fuzzy extractor as T_{em}, T_h, $T_{e/d}$, and T_f. We used a cryptography library called MIRACL to measure all operations in these protocols. We conducted experiments in different environments, considering the computing performance of vehicles and fog nodes. First, we conducted an experiment on a desktop equipped with an i7-4790 intel CPU, 16 GB of RAM, and a Linux Ubuntu 20.04-desktop-amd64 operating system to reflect the high computing performance of the fog nodes. Moreover, we conducted the same experiment on a Raspberry PI 3B with an ARM Cortex-A53 and 1 GB of RAM to reflect the low computing performance of vehicles. We summarized the execution time for each operation in Table 4. We configured the PUF response produced by a fuzzy extractor to ensure noise resilience, assuming that the execution time is the same as that for ECC scalar multiplication. We investigated the computational costs deriving from all operations performed during the authentication phase of these protocols. In Ma et al.'s protocol [17], a vehicle performed three ECC scalar multiplications and four hash operations, so we calculated the computational cost of the vehicle as $3T_{em} + 4T_h = 4.479$ ms. A fog node performed four ECC scalar multiplications and four hash operations, so we calculated the computational cost of the fog node as $4T_{em} + 4T_h = 5.968$ ms. The cloud server performed eight ECC scalar multiplications and nine hash operations, so we calculated the computational cost of the cloud server as $8T_{em} + 9T_h = 11.939$ ms. Thus, the total cost of vehicle, fog node, and cloud server is $15T_{em} + 17T_h = 22.386$ ms.

Table 4. Execution time for each operation.

Notations	Descriptions	Desktop	Raspberry PI
T_{em}	ECC scalar multiplication	1.489 ms	2.579 ms
T_h	Hash operation	0.003 ms	0.021 ms
$T_{e/d}$	Symmetric cryptography operation	0.001 ms	0.013 ms
T_f	Fuzzy extractor	1.489 ms	2.579 ms

In Eftekhari et al.'s protocol [18], a vehicle performed three ECC scalar multiplication operations and fourteen hash operations, so we calculated the computational cost of the vehicle as $3T_{em} + 14T_h = 4.509$ ms. A fog node performed three ECC scalar multiplications and sixteen hash operations, so we calculated the computational cost of the fog node as $3T_{em} + 16T_h = 4.515$ ms. The cloud server performed three ECC scalar multiplications and seventeen hash operations, so we calculated the computational cost of the cloud server as $3T_{em} + 17T_h = 4.518$ ms. Thus, the total cost of vehicle, fog node, and cloud server is $9T_{em} + 47T_h = 13.542$ ms.

In Kumar et al.'s protocol [19], a vehicle performed five ECC scalar multiplications, eleven hash operations, and one symmetric cryptography operation, so we calculated the computational cost of the vehicle as $5T_{em} + 11T_h + 1T_{e/d} = 7.479$ ms. A fog node performed five ECC scalar multiplications, ten hash operations, and two symmetric cryptography operations, so we calculated the computational cost of the fog node as $5T_{em} + 10T_h + 2T_{e/d} = 7.477$ ms. The cloud server performed two ECC scalar multiplications, three hash operations, and three symmetric cryptography operations, so we calculated the computational cost of the cloud server as $2T_{em} + 3T_h + 3T_{e/d} = 2.99$ ms. Thus, the total cost of vehicle, fog node, and cloud server is $12T_{em} + 24T_h + 6T_{e/d} = 17.946$ ms.

In Wu et al.'s protocol [20], a vehicle performed two ECC scalar multiplications and eight hash operations, so we calculated the computational cost of the vehicle as $2T_{em} + 8T_h = 3.002$ ms. A fog node performed four ECC scalar multiplications and five hash operations, so we calculated the computational cost of the fog node as $4T_{em} + 5T_h = 5.971$ ms. The cloud server performed three ECC scalar multiplications and thirteen hash operations, so we calculated the computational cost of the cloud server as $3T_{em} + 13T_h = 4.506$ ms. Thus, the total cost of vehicle, fog node, and cloud server is $9T_{em} + 26T_h = 13.479$ ms.

In Awais et al.'s protocol [8], a vehicle performed three ECC scalar multiplications and six hash operations, so we calculated the computational cost of the vehicle as $3T_{em} + 6T_h = 3.002$ ms. A fog node performed four ECC scalar multiplications and four hash operations, so we calculated the computational cost of the fog node as $4T_{em} + 4T_h = 5.971$ ms. The cloud server performed four ECC scalar multiplications and nine hash operations, so we calculated the computational cost of the cloud server as $4T_{em} + 9T_h = 4.506$ ms. Thus, the total cost of vehicle, fog node, and cloud server is $11T_{em} + 19T_h = 16.436$ ms.

In Hedge et al.'s protocol [21], a smart device performed three ECC scalar multiplications and thirteen hash operations, so we calculated the computational cost of the vehicle as $3T_{em} + 13T_h = 4.506$ ms. A fog node performed five ECC scalar multiplications and ten hash operations, so we calculated the computational cost of the fog node as $5T_{em} + 10T_h = 7.475$ ms. The cloud server performed four ECC scalar multiplications and six hash operations, so we calculated the computational cost of the cloud server as $4T_{em} + 6T_h = 5.974$ ms. Thus, the total cost of smart device, fog node and cloud server is $12T_{em} + 29T_h = 17.955$ ms.

In Awais et al.'s protocol [22], we calculated the computational costs by combining those of the fog nodes and the RSUs. A vehicle performed three ECC scalar multiplications and three hash operations, so we calculated the computational cost of the vehicle as $3T_{em} + 3T_h = 4.476$ ms. A fog node performed five ECC scalar multiplications and five hash operations, so we calculated the computational cost of the fog node as $5T_{em} + 5T_h = 7.46$ ms. The cloud server performed six ECC scalar multiplications and ten hash operations, so we calculated the computational cost of the cloud server as $6T_{em} + 10T_h = 8.964$ ms. Thus, the total cost of vehicle, fog node and cloud server is $14T_{em} + 18T_h = 20.9$ ms.

In our proposed protocol, a vehicle performed three ECC scalar multiplications and fifteen hash operations, so we calculated the computational cost of the vehicle as $3T_{em} + 15T_h = 4.512$ ms. A fog node performed two ECC scalar multiplications and ten hash operations, so we calculated the computational cost of the fog node as $2T_{em} + 10T_h = 3.008$ ms. The cloud server performed three ECC scalar multiplications and twelve hash operations, so we calculated the computational cost of the cloud server as $3T_{em} + 12T_h = 4.503$ ms. Thus, the total cost of vehicle, fog node, and cloud server is $8T_{em} + 37T_h = 12.023$ ms.

We show the comparison results of computational cost of our proposed protocol and other related protocols in Table 5 and Figure 6. The results show that our proposed protocol

has a lower total computational cost, especially on the fog node side, than related protocols. Therefore, we can state that our proposed protocol has relatively higher computational efficiency than other related protocols.

Table 5. Comparison of computational costs.

Protocol	Vehicle User	Fog Node	Cloud Server	Total
[17]	4.479 ms	5.968 ms	11.939 ms	22.386 ms
[18]	4.509 ms	4.515 ms	4.518 ms	13.542 ms
[19]	7.479 ms	7.477 ms	2.99 ms	17.946 ms
[20]	3.002 ms	5.971 ms	4.506 ms	13.479 ms
[8]	4.485 ms	5.968 ms	5.983 ms	16.436 ms
[21]	4.506 ms	7.475 ms	5.974 ms	17.955 ms
[22]	4.476 ms	7.46 ms	8.964 ms	20.9 ms
Proposed	4.512 ms	3.008 ms	4.503 ms	12.023 ms

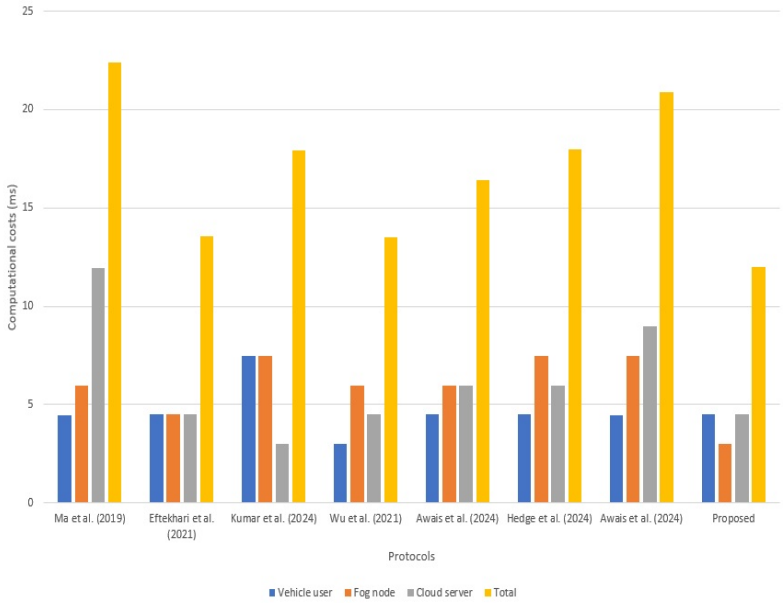

Figure 6. Visualization of computational costs comparison [8,17–22].

6.2. Communication Cost Analysis

This subsection compares the communication cost of our proposed protocol with [8,17–22]. We denoted that the ECC point, hash output, random nonce, identity, and timestamp were 320, 256, 256, 128, and 32 bits, respectively. In Ma et al.'s protocol [17], a vehicle transmitted $\{AID_{U_i}, T_{U_i}, R_1, \alpha\}$, so we calculated the communication cost of the vehicle as 864 bits. A fog node transmitted $\{AID_{U_i}, AID_{FN_j}, T_{U_i}, T_{FN_j}, R_1, R_2, \hat{R}_2, \alpha, \beta\}$ and $\{R_2, R_3, \hat{R}_3', T_{CS}, \bar{\gamma}\}$, so we calculated the communication cost of the fog node as 3296 bits. The cloud server transmitted $\{R_3, \hat{R}_3, \hat{R}_3', T_{CS}, \gamma, \bar{\gamma}\}$, so we calculated the communication cost of the cloud server as 1504 bits. Thus, the total cost of vehicle, fog node, and cloud server is 5664 bits.

In Eftekhari et al.'s protocol [18], a vehicle transmitted $\{RID_{DR}, X_{VE}, y_{VE}, h_{VE}^{CS}, T\}$, so we calculated the communication cost of the vehicle as 1120 bits. A fog node transmitted $\{RID_{FS}, RID_{DR}, X_{FS}, X_{VE}, y_{FS}, y_{VE}, h_{FS}^{CS}, T\}$ and $\{mRID_{CS}^{DR_{new}}, X_{FS}, X_{CS}, h_{FS}^{VE}\}$, so we calculated the communication cost of the fog node as 3104 bits. The cloud server transmitted $\{mRID_{CS}^{DR_{new}}, mRID_{CS}^{FS_{new}}, X_{CS}, h_{CS}^{FS}, h_{CS}^{VE}\}$, so we calculated the communication cost of the cloud server as 1344 bits. Thus, the total cost of vehicle, fog node, and cloud server is 5568 bits.

In Kumar et al.'s protocol [19], a vehicle transmitted $M_1 = \{PK_V, PIDV_i, V_1, T_1\}$ and $M_5 = \{PK_{V_1}, V_5, T_5\}$, so we calculated the communication cost of the vehicle as 1600 bits. A fog node transmitted $M_2 = \{M_1, PK_F, PIDFN_j, V_2, T_2\}$ and $M_4 = \{PK_{F_1}, V_4, T_4\}$, so we calculated the communication cost of the fog node as 2592 bits. The cloud server transmitted $M_3 = \{C_3, V_3, T_3\}$, so we calculated the communication cost of the cloud server as 544 bits. Thus, the total cost of vehicle, fog node, and cloud server is 4736 bits.

In Wu et al.'s protocol [20], a vehicle transmitted $M_1 = \{PID_i, N_i, B_2, B_3, T_v\}$, so we calculated the communication cost of the vehicle as 992 bits. A fog node transmitted $M_2 = \{M_1, PFSID_j, N_j, N_{ij}, B_5, B_6, T_f\}$ and $M_4 = \{K_{ij}, V_1, N_{jc}, T_c, T_{f2}\}$, so we calculated the communication cost of the fog node as 3200 bits. The cloud server transmitted $M_3 = \{K_{ij}, V_1, V_2, N_c, N_{ic}, T_c\}$, so we calculated the communication cost of the cloud server as 1440 bits. Thus, the total cost of vehicle, fog node, and cloud server is 5632 bits.

In Awais et al.'s protocol [8], a vehicle transmitted $D_1 = \{R_1, RID_i, Q_i\}$, so we calculated the communication cost of the vehicle as 832 bits. A fog node transmitted $D_2 = \{D_1, R_2, \hat{R}_2, RID_j, L_j\}$ and $D_4 = \{R_2, R_3, X_i, Auth_i\}$, so we calculated the communication cost of the fog node as 3136 bits. The cloud server transmitted $D_3 = \{R_3, Y_j, X_i, Auth_i, Auth_j\}$, so we calculated the communication cost of the cloud server as 1344 bits. Thus, the total cost of vehicle, fog node, and cloud server is 5312 bits.

In Hedge et al.'s protocol [21], a smart device transmitted $\{CID_s, RV_2, C_{sm}, T_1\}$, so we calculated the communication cost of the vehicle as 864 bits. A fog node transmitted $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$ and $\{F_{sm}, T_4, FCSUID_i, T_3, CV_2\}$, so we calculated the communication cost of the fog node as 3136 bits. The cloud server transmitted $\{CV_2, T_3\}$, so we calculated the communication cost of the cloud server as 352 bits. Thus, the total cost of vehicle, fog node, and cloud server is 4352 bits.

In Awais et al.'s protocol [22], we calculated the computational costs by combining those of the fog nodes and the RSUs. A vehicle transmitted $M_1 = \{TID_{U_i}, R_1, \alpha\}$, so we calculated the communication cost of the vehicle as 704 bits. A fog node transmitted $M_2 = \{M_1, TID_{RSU_k}, R_2, \beta\}$, $M_3 = \{M_2, TID_{FN_j}, R_3, \gamma\}$, $M_5 = \{R_{10}, X_k\}$, and $M_6 = \{R_9, X_i\}$, so we calculated the communication cost of the fog node as 4672 bits. The cloud server transmitted $M_4 = \{R_7, R_8, R_9, X_j\}$, so we calculated the communication cost of the cloud server as 1344 bits. Thus, the total cost of vehicle, fog node, and cloud server is 6592 bits.

In our proposed protocol, a vehicle transmitted $D_1 = \{R_1, RID_i, Q_i, ID_j\}$, so we calculated the communication cost of the vehicle as 960 bits. A fog node transmitted $D_2 = \{R_1, RID_i, R_2, r_2R_1, L_j, ID_j\}$ and $D_4 = \{K_{ij}, r_2R_3, Auth_i\}$, so we calculated the communication cost of the fog node as 2432 bits. The cloud server transmitted $D_3 = \{K_{ij}, R_3, r_3R_1, Auth_j\}$, so we calculated the communication cost of the cloud server as 1152 bits. Thus, the total cost of vehicle, fog node, and cloud server is 4544 bits.

We show the comparison results of communication cost for our proposed protocol and other related protocols in Table 6 and Figure 7. In Table 6 and Figure 7, the results show that our proposed protocol has the lowest total communicational cost among other related protocols. Therefore, we can state that our proposed protocol has relatively higher communication efficiency than other related protocols.

Table 6. Comparison of communication costs.

Protocols	Communication Costs
[17]	5664 bits
[18]	5568 bits
[19]	4736 bits
[20]	5632 bits
[8]	5312 bits
[21]	4352 bits
[22]	6592 bits
Proposed	4544 bits

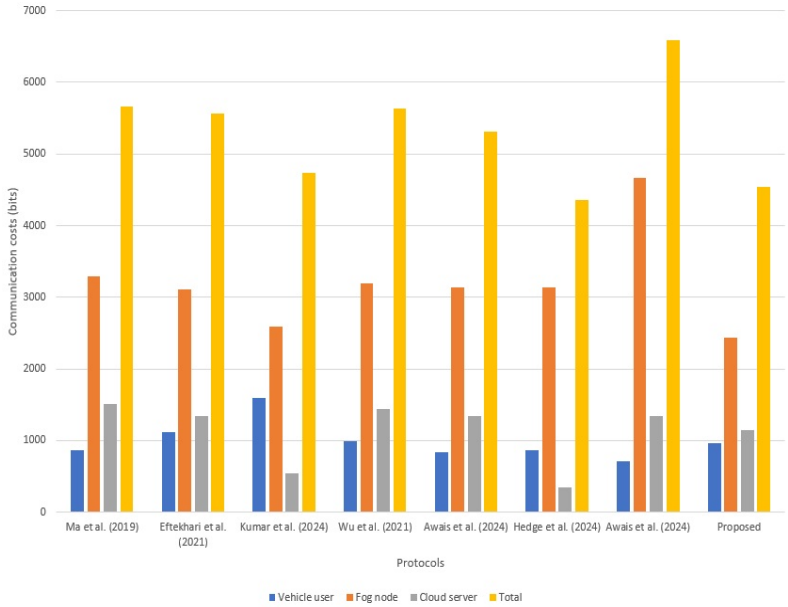

Figure 7. Visualization of communication costs comparison [8,17–22].

6.3. Security Features

We compared the security features of the proposed protocol with those of related protocols [8,17–22]. We considered (SF1) "preservation of anonymity", (SF2) "preservation of untraceability", (SF3) "preservation of perfect forward secrecy", (SF4) "resistance to stolen verifier attack", (SF5) "resistance to stolen smart card attack", (SF6) "resistance to session key disclosure attack", (SF7) "resistance to replay attack", (SF8) "resistance to MITM attack", (SF9) "resistance to ephemeral secret leakage attack", and (SF10) "resistance to privileged insider attack". Table 7 summarizes the comparison of security features. The results show that the proposed protocol has superior security than other related protocols in fog-based VANET environments.

Table 7. Comparison of security features.

Security Features	[17]	[18]	[19]	[20]	[8]	[21]	[22]	Proposed
SF1	O	O	O	O	O	O	O	O
SF2	O	O	O	O	O	O	O	O
SF3	O	O	O	O	X	-	O	O
SF4	O	O	O	-	O	-	O	O
SF5	O	O	-	-	X	O	-	O
SF6	O	-	-	-	O	-	O	O
SF7	O	O	O	O	O	O	O	O
SF8	O	O	O	O	O	O	O	O
SF9	X	O	O	O	X	O	O	O
SF10	X	-	-	-	-	O	-	O

-: Not considered. X: Insecure. O: Secure.

7. Conclusions

In this study, we proposed a lightweight and robust authentication protocol for securing fog-based VANETs. Considering the features of fog-based VANETs, we used the ECC system and fuzzy verifier to establish a session key securely and efficiently for vehicle-to-infrastructure communication. The proposed protocol provides perfect forward secrecy and resists various attacks, such as trace and ephemeral secret leakage attacks. Furthermore, we conducted informal and formal security analyses to demonstrate the efficiency and security robustness of our protocol. The informal security analysis demonstrated that our protocol satisfies security requirements and the formal security analysis using BAN logic, the AVISPA simulation tool, and the RoR model, demonstrating that our protocol offers mutual authentication and session key security. Finally, we compared the performance of our protocol with that of other related protocols to evaluate its efficiency. The results demonstrated that our protocol outperformed the compared protocols in computational and communication cost. In the future, we plan to assess our protocol's practical issues through simulations that consider actual VANET conditions. Moreover, future research will aim to expand this study to enable secure communication under various network conditions. Additionally, we plan to introduce outsourcing computing methods to lower the computational costs for vehicles, and these improvements will expand the potential applications of intelligent vehicle systems.

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

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (RS-2024-00450915).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Al-Sultan, S.; Al-Doori, M.M.; Al-Bayatti, A.H.; Zedan, H.A comprehensive survey on vehicular ad hoc network. *J. Netw. Comput. Appl.* **2014**, *37*, 380–392.
2. Prajapat, S.; Gautam, D.; Kumar, P.; Jangirala, S.; Das, A.K.; Park, Y.; Lorenz, P. Secure lattice-based aggregate signature scheme for vehicular Ad Hoc networks. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12370–12384.
3. Yousefi, S.; Mousavi, M.S.; Fathy, M. Vehicular ad hoc networks (VANETs): Challenges and perspectives. In Proceedings of the 2006 6th International Conference on ITS Telecommunications, Chengdu, China, 21–23 June 2006; pp. 761–766.
4. Yi, S.; Li, C.; Li, Q. A Survey of Fog Computing. In Proceedings of the 2015 Workshop on Mobile Big Data-Mobidata '15, Hangzhou, China, 21 June 2015; pp. 37–42.
5. Sookhak, M.; Yu, F.R.; He, Y.; Talebian, H.; Safa, N.S.; Zhao, N.; Khan, M.K.; Kumar, N. Fog Vehicular Computing: Augmentation of Fog Computing Using Vehicular Cloud Computing. *IEEE Veh. Technol. Mag.* **2017**, *12*, 55–64.
6. Huang, C.; Lu, R.; Choo, K.K.R. Vehicular fog computing: Architecture, use case, and security and forensic challenges. *IEEE Commun. Mag.* **2017**, *55*, 105–111.
7. Stojmenovic, I.; Wen, S.; Huang, X.; Luan, H. An Overview of Fog Computing and Its Security Issues. *Concurr. Comput. Pract. Exp.* **2016**, *28*, 2991–3005.
8. Awais, S.; Yucheng, W.; Mahmood, K.; Muhammad, H.; Badar, S.; Kharel, R.; Das, A. Provably secure fog-based authentication protocol for VANETs. *Comput. Netw.* **2024**, *246*, 110391.
9. Hou, X.; Li, Y.; Chen, M.; Wu, D.; Jin, D.; Chen, S. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures. *IEEE Trans. Veh. Technol.* **2016**, *65*, 3860–3873.
10. Peixoto, M.L.M; Maia, A.H.O.; Mota, E.; Rangel, E.; Costa, D.G.; Turgut, D.; Villas L.A. A traffic data clustering framework based on fog computing for VANETs. *Veh. Commun.* **2021**, *31*, 100370.
11. Pereira, J.; Ricardo, L.; Luís, M.; Senna, C.; Sargento, S. Assessing the reliability of fog computing for smart mobility applications in VANETs. *Future Gener. Comput. Syst.* **2019**, *94*, 317–332.
12. Farooqi, A.M.; Alam, M.A.; Hassan, S.I.; Idrees, S.M. A Fog Computing Model for VANET to Reduce Latency and Delay Using 5G Network in Smart City Transportation. *Appl. Sci.* **2022**, *12*, 2083.
13. AlHamid, H.A.; Rahman, S.M.M.; Hossain, M.S.; Almogren, A.; Alamri, A. A Security Model for Preserving the Privacy of Medical Big Data in a Healthcare Cloud Using a Fog Computing Facility with Pairing-Based Cryptography. *IEEE Access* **2017**, *5*, 22313–22328.
14. Jia, X.; He, D.; Kumar, N.; Choo, K.K.R. Authenticated key agreement scheme for fog-driven IoT healthcare system. *Wirel. Netw.* **2019**, *25*, 4737–4750.
15. Okay, F.Y.; Ozdemir, S. A secure data aggregation protocol for fog computing based smart grids. In Proceedings of the 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018), Doha, Qatar, 10–12 April 2018; pp. 1–6.
16. Lyu, L.; Nandakumar, K.; Rubinstein, B.; Jin, J.; Bedo, J.; Palaniswami, M. PPFA: Privacy preserving fog-enabled aggregation in smart grid. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3733–3744.
17. Ma, M.; He, D.; Wang, H.; Kumar, N.; Choo, K.K.R. An Efficient and Provably-Secure Authenticated Key Agreement Protocol for Fog-Based Vehicular Ad-Hoc Networks. *IEEE Internet Things J.* **2019**, *6*, 8065–8075.
18. Eftekhari, S.A.; Nikooghadam, M.; Rafighi, M. Security-enhanced three-party pairwise secret key agreement protocol for fog-based vehicular ad-hoc communications. *Veh. Commun.* **2021**, *28*, 100306.
19. Kumar, P.; Om, H. Multi-TA model-based conditional privacy-preserving authentication protocol for fog-enabled VANET. *Veh. Commun.* **2024**, *47*, 100785.
20. Wu, T.Y.; Lee, Z.; Yang, L.; Luo, J.N.; Tso, R. Provably secure authentication key exchange scheme using fog nodes in vehicular ad hoc networks. *J. Supercomput.* **2021**, *77*, 6992–7020.
21. Hegde, M.; Rao, R.R.; Bhat, R. Design of an Efficient and Secure Authentication Scheme for Cloud-Fog-Device Framework Using Key Agreement and Management. *IEEE Access* **2024**, *12*, 78173–78192.
22. Awais, S.; Yucheng, W.; Mahmood, K.; Alenazi, M.; Bashir, A.; Das, A.; Lorenz, P. Provably secure and lightweight authentication and key agreement protocol for fog-based vehicular ad-hoc networks. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 21107–21116.
23. Hankerson, D.; Menezes, A.J.; Vanstone, S. *Guide to Elliptic Curve Cryptography;* Springer: Berlin/Heidelberg, Germany, 2004.
24. Dolev, D.; Yao, A.C.-C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–207.
25. Ryu, J.; Son, S.; Lee, J.; Park, Y.; Park, Y. Design of secure mutual authentication scheme for metaverse environments using blockchain. *IEEE Access* **2022**, *10*, 98944–98958.
26. Yu, S.; Lee, J.; Sutrala, A.K.; Das, A.K.; Park, Y. LAKA-UAV: Lightweight authentication and key agreement scheme for cloud-assisted Unmanned Aerial Vehicle using blockchain in flying ad hoc networks. *Comput. Netw.* **2023**, *224*, 109612.

27. Canetti, R.; Krawczyk, H. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *EUROCRYPT 2001: Advances in Cryptology, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, 15–19 May 2001;* Pfitzmann, B., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2045.
28. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.; Park, Y. AKM-IoV: Authenticated Key Management Protocol in Fog Computing-Based Internet of Vehicles Deployment. *IEEE Internet Things J.* **2019**, *6*, 8804–8817.
29. Das, A.K.; Wazid, M.; Yannam, A.R.; Rodrigues, J.J.; Park, Y. Provably secure ECC-based device access control and key agreement protocol for IOT environment. *IEEE Access* **2019**, *7*, 55382–55397.
30. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
31. Yu, S.; Park, Y. A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions. *IEEE Internet Things J.* **2022**, *9*, 20214–20228.
32. Son, S.; Kwon, D.; Lee, S.; Jeon, Y.; Das, A.K.; Park, Y. Design of secure and lightweight authentication scheme for UAV-enabled intelligent transportation systems using blockchain and PUF. *IEEE Access* **2023**, *11*, 60240–60253.
33. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *Proc. R. Soc. Lond. A Math. Phys. Sci.* **1989**, *426*, 233–271.
34. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In Proceedings of the 8th International Workshop on Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005; Volume 3386, pp. 65–84.
35. 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.
36. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791.
37. Vigano, L. Automated security protocol analysis with the AVISPA tool. *Electron. Notes Theor. Comput. Sci.* **2006**, *155*, 61–86.