Hindawi
Mobile Information Systems
Volume 2021, Article ID 3277113, 14 pages
https://doi.org/10.1155/2021/3277113

Research Article

A Lightweight Authenticated Key Agreement Protocol Using Fog Nodes in Social Internet of Vehicles

Tsu-Yang Wu,[1] **Xinglan Guo**,[1] **Lei Yang**,[1] **Qian Meng**,[1] **and Chien-Ming Chen**[2]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China*

Correspondence should be addressed to Chien-Ming Chen; chienmingchen@ieee.org

Received 2 September 2021; Accepted 27 October 2021; Published 17 November 2021

Academic Editor: Ke Gu

Recently, there has been rapid growth in the Internet of things, the Internet of vehicles, fog computing, and social Internet of vehicles (SIoV), which can generate large amounts of real-time data. Now, researchers have begun applying fog computing to the SIoV to reduce the computing pressure on cloud servers. However, there are still security challenges in SIoV. In this paper, we propose a lightweight and authenticated key agreement protocol based on fog nodes in SIoV. The protocol completes the mutual authentication between entities and generates the session key for subsequent communication. Through a formal analysis of the Burrows–Abadi–Needham (BAN) logic, real-oracle random (ROR) model, and ProVerif, the security, validity, and correctness of the proposed protocol are demonstrated. In addition, informal security analysis shows that our proposed protocol can resist known security attacks. We also evaluate the performance of the proposed protocol and show that it achieves better performance in terms of computing power and communication cost.

1. Introduction

With the popularization and development of the world wide web, the Internet of things (IoT) [1–3], which is a network of Internet extension and expansion, has emerged. With the continuous development of IoT applications, a "social network of intelligent objects" called social Internet of things (SIoT) [4] has been formed. Internet of vehicles (IoV) [5] is an extension of the concept of SIoT. IoV can realize network connections between the vehicle and vehicle (V2V), vehicle and infrastructure (V2I), and vehicle and pedestrian (V2P) and collect and share the key road information. With the rapid development of network and sensor technology, social connection in urban transportation systems is necessary, so social Internet of vehicles (SIoV) is produced [6–8]. SIoV is an application of SIoT in the field of vehicles and is a combination of vehicular ad hoc networks (VANET) and mobile networks, and it can generate a large amount of real-time data. In SIoV, intelligent vehicles can establish social relationships with other objects and form a specific social network.

For cloud computing processing of road real-time data, there are some problems associated with network delays, transmission efficiency, and others. Because the distance between the cloud computing server and vehicles is far, and the number of vehicles is increasing, the cloud server needs to process more real-time data, which increases the computing burden. Therefore, researchers have introduced fog computing to reduce the computational burden on cloud servers. The data, processing, and application of fog computing are stored on scattered and weak devices, almost outside the cloud, so the computing power is not strong. It can help the cloud server process some data that are not necessary or urgent at that moment. If it encounters data that it cannot process, it reports to the cloud server. Fog nodes can detect unsafe driving behavior in time, issue early warnings for the behavior, and provide the corresponding punishment when necessary. The application of fog node in IoT and IoV environments was mentioned in the articles [9–13]. In 2016, Azimi et al. [11] proposed a medical warning system in IoT based on fog computing. In 2019, Ismail et al. [12] proposed an implication of fog computing on the IoT.

In 2019, Ma et al. [10] proposed a protocol for fog-based IoV networks, which realized authenticated key agreement. In 2021, Eftekhari et al. [9] proposed a pairwise secret key agreement protocol using fog-based IoV, which was a three-part authentication protocol. The SIoV typical architecture based on fog nodes is shown in Figure 1.

However, in the SIoV environment based on fog nodes, there are still great risks related to security issues. For example, it is very challenging to ensure the confidentiality and privacy of data transmission based on ensuring the security of devices deployed on the network edge. The data transmitted through the public channel usually includes sensitive information such as the personal information of vehicle users, which needs to be kept secret. Recently, Ahmed et al. [6] researched a key agreement protocol for V2G in the SIoV environment, which was a two-party authentication protocol. The protocol [6] was based on an elliptic-curve (ECC) point multiplication and had a large computational cost. This shortcoming leads us to propose a more effective protocol.

We propose a lightweight and authenticated key agreement protocol based on three parties using fog nodes in an SIoV environment. In this protocol, vehicles and fog nodes authenticate each other with the help of a cloud server (CS) and establish a secure session key. Owing to the weak computing power of fog nodes, our protocol only uses lightweight primitives, such as hash function and XOR operation. Through formal analysis of the Burrows–Abadi–Needham (BAN) logic, real-oracle random (ROR) model, and ProVerif, the security, validity, and correctness of the proposed protocol are demonstrated. In addition, informal security analysis shows that our proposed protocol can resist known security attacks. We also evaluate the performance of the proposed protocol and show that it has better performance in terms of computing power and communication cost.

The rest of the paper is structured as follows: in Section 2, we review recent research results. The details of our proposed agreement are in Section 3. In Section 4, we use BAN, ROR, and ProVerif to verify the security, validity, and correctness of the proposed protocol. In addition, we conduct an informal security analysis. In Section 5, we compare our method with other protocols in terms of performance and security. Finally, we summarize this paper in Section 6.

2. Related Work

IoV is an open network environment, so this feature may threaten the identity information and relevant sensitive data of vehicle users. For many years, researchers have proposed many protocols to protect the privacy of vehicle users in IoV environments. In 2006, Raya et al. [14] proposed a vehicle communication protocol that stored multiple public and private key pairs and protected the privacy of vehicle users through the certificates stored in OBU. However, in 2008, Lu et al. [15] determined that the protocol [14] had high computing and storage cost because the key was changing at times and proposed a privacy protection protocol for vehicle

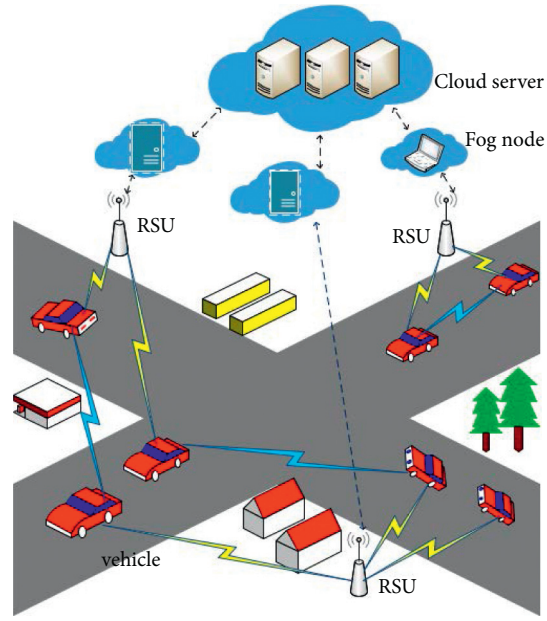

Figure 1: The SIoV architecture based on fog node.

communication. That same year, Zhang et al. [16] proposed an identity verification protocol for *IoV*. The protocol [16] realized privacy protection by the tamper-proof device to generate a random pseudoidentity. In 2020, Cui et al. [17] researched a privacy-preserving scheme. The protocol [17] was based on edge computing and used lightweight primitives, such as elliptic-curve cryptography, instead of bilinear pairing-based primitives with high computational cost. Later, Hu et al. [18] proposed a privacy-preserving authentication scheme for IoV.

The protocols proposed by some researchers have high computing power. In 2014, Li et al. [19] proposed a protocol that provided PKC-based privacy protection for IoV and claimed that their protocol could resist replay and stolen smart card attacks. However, Amit et al. [20] revealed that Li et al.'s protocol [19] was susceptible to key compromise impersonation attacks and could not provide user anonymity. To reduce high computing cost caused by the use of PCK in the above protocol, the Trust-Extended Authentication Mechanism (TEAM) protocol was proposed [21]. In 2016, Kumari et al. [22] proposed an authentication protocol that also used TEAM. In 2017, Ying and Nayak [23] proposed an effective and lightweight protocol for an IoV environment, which could provide user anonymity. Chen et al. [5] demonstrated that [23] was vulnerable to replay and offline identity guessing attacks. Therefore, to solve the vulnerability of Ying and Nayak's protocol [23], Chen et al. [5] proposed a secure authentication scheme for IoV. However, the protocol [5] stored extensive data in the database, so it had high storage cost. In the same year, Mohit et al. [24] proposed an efficient authentication protocol for vehicular systems and deemed their protocol safe. However, Yu et al. [25] pointed out that the protocol [24] of Mohit et al. was susceptible to impersonation attacks and could not provide anonymity, traceability, and mutual authentication. Then, Yu et al. [25] proposed an authenticated protocol in

vehicular communications. In 2020, Sadri et al. [26] demonstrated that Yu et al.'s protocol [25] was susceptible to sensor capture attacks and impersonation attacks and could not provide traceability. Additionally, Sadri and Rajabzadeh Asaar [26] proposed a protocol in the IoV environment, which was based on lightweight primitives. In 2021, Wu et al. [27] proposed a protocol in IoV, and the protocol realized authentication key exchange (AKE).

There are increasingly more vehicles in the IoV environment, and data processing and transmission have become an inevitable challenge. Therefore, researchers began to apply cloud computing to IoV to solve the problem of processing a large amount of data to improve authentication efficiency. In an IoV environment, an authentication scheme based on cloud computing had been widely mentioned and applied in articles [28–31]. For an environment using cloud computing, problems such as network delay and transmission efficiency would exist, and the cloud server would need to process more data, which would increase the computing burden of the cloud server. Therefore, researchers have begun to introduce fog nodes for fog computing to share the pressure of cloud servers. In these papers [10–13], fog computing technology was applied. Ma et al.'s protocol [10] applied fog computing to IoV and proposed an authenticated key agreement protocol. They claimed that the protocol [10] was secure and efficient, but Eftekhari et al. [9] pointed out that Ma et al.'s protocol [10] was vulnerable to internal attacks, stolen smart card attacks, and known session-specific temporary information attacks. Therefore, Eftekhari et al. [9] proposed a more efficient authentication protocol. In 2021, Wu et al. [32] proposed a secure scheme using fog nodes in IoV, and the protocol realized AKE. In the same year, Maria et al. [33] proposed a blockchain-based anonymous authentication scheme, which used bilinear pairing. Some important related works are summarized in Table 1.

3. The Proposed Protocol

In this part, we introduce a lightweight and authenticated key agreement protocol using fog nodes in SIoV. Our protocol is based on the architecture of Figure 1. The protocol includes three entities: vehicle V_i, fog node FN_j, and CS. The symbols used in the protocol are shown in Table 2. The protocol has three phases: vehicle registration phase, fog node registration phase, and login authentication phase.

3.1. V_i Registration Phase. In the V_i registration phase, V_i registers with CS. The phase is shown in Figure 2, and the specific steps are as follows:

(1) First, V_i selects its identity ID_i, password PSW_i, and a random number r_i, calculates its pseudoidentity $\mathrm{PID}_i = h(\mathrm{ID}_i \| r_i)$, and then transmits the PID_i to CS through the secure channel.

(2) After receiving the message from V_i, CS calculates the value of $\mathrm{HID}_i = h(\mathrm{PID}_i \| K_{\mathrm{CS}})$, initializes the value of K_V to 0, and stores $\{\mathrm{PID}_i, K_V\}$ in its database. Finally, CS sends $\{\mathrm{HID}_i, K_V\}$ to V_i.

(3) After receiving the message from CS, V_i calculates the value $\alpha_i = \mathrm{HID}_i \oplus h(\mathrm{PSW}_i \| r_i)$, $P_i = h(\mathrm{ID}_i \| \mathrm{PSW}_i \| r_i)$, replaces HID_i with the value of α_i, and stores the $\{\alpha_i, P_i, r_i, K_V\}$ in its smart card.

3.2. F_j Registration Phase. In FN_j registration phase, FN_j registers with CS. The phase is shown in Figure 3, and the specific steps are as follows:

(1) First, FN_j selects its identity FID_j and a random number r_j, calculates its pseudoidentity $\mathrm{PFID}_j = h(\mathrm{FID}_j \| r_j)$, and then transmits $\{\mathrm{PFID}_j, \mathrm{FID}_j\}$ to CS through the secure channel.

(2) After receiving the message from FN_j, CS first selects a random number R_j, calculates the value of $N_j = h(\mathrm{FID}_j \| \mathrm{ID}_{\mathrm{CS}}) \oplus R_j$, $K_{\mathrm{FN}} = h(\mathrm{PFID}_j \| K_{\mathrm{CS}})$, $\mathrm{HID}_j = h(\mathrm{FID}_j \| K_{\mathrm{CS}})$, and stores $\{\mathrm{PFID}_j, K_{\mathrm{FN}}, \mathrm{FID}_j\}$ in its database. Finally, CS sends $\{K_{\mathrm{FN}}, \mathrm{HID}_j, N_j, \mathrm{ID}_{\mathrm{CS}}\}$ to FN_j.

(3) After receiving the message from CS, FN_j calculates the value $R_j = h(\mathrm{FID}_j \| \mathrm{ID}_{\mathrm{CS}}) \oplus N_j$, $\beta_j = \mathrm{HID}_j \oplus h(R_j \| r_j)$, and stores the $\{K_{\mathrm{FN}}, \beta_j, r_j, N_j\}$ in its database.

3.3. Login and Authentication Phase. In the login and authentication phase, V_i, FN_j, and CS realize authentication and establish session key SK. This phase is shown in Figure 4, and the specific steps are as follows:

(1) First, V_i inserts the smart card into the reader terminal, inputs its identity ID_i, password PSW_i, calculates the login authentication value $P_i^* = h(\mathrm{ID}_i \| \mathrm{PSW}_i \| r_i)$, and then compares $P_i^* \stackrel{?}{=} P_i$. If equal, V_i logs in successfully. Otherwise, the login fails. After successful login, V_i selects a random number N_1 and calculates $A_1 = h(\mathrm{ID}_i \| r_i) \oplus N_1$, $\mathrm{HID}_i = \alpha_i \oplus h(\mathrm{PSW}_i \| r_i)$, $V_1 = h(\mathrm{HID}_i \| K_V) \oplus N_1$. Finally, V_i sends the login request $M_1 = \{A_1, V_1, \mathrm{ID}_{\mathrm{CS}}, \mathrm{PID}_i\}$ to FN_j through the common channel.

(2) After receiving the message M_1 from V_i, FN_j first selects a random number N_2 and then calculates $A_2 = h(A_1 \| K_{\mathrm{FN}} \| \mathrm{HID}_j) \oplus N_2$, $V_2 = h(A_2 \| K_{\mathrm{FN}} \| V_1)$, and finally FN_j transmits the message $M_2 = \{\mathrm{PID}_i, \mathrm{PFID}_j, A_2, V_1, V_2\}$ to CS.

(3) After receiving message M_2 from FN_j, CS first indexes K_{FN} according to FPID_j, then calculates $\mathrm{HID}_i = h(\mathrm{PID}_i \| K_{\mathrm{CS}})$, $N_1 = h(\mathrm{HID}_i \| K_V) \oplus V_1$, $V_1^* = h(\mathrm{HID}_i \| K_V) \oplus N_1$, and compares $V_1^* \stackrel{?}{=} V_1$. If it is equal, CS believes that V_i is legal. Otherwise, the authentication process is terminated. CS calculates $V_2^* = h(A_2 \| K_{\mathrm{FN}} \| V_1)$ and compares $V_2^* \stackrel{?}{=} V_2$. If it is equal, it means that CS believes that FN_j is legal. Otherwise, the authentication process is terminated. After authenticating V_i and FN_j, CS calculates $A_1 = N_1 \oplus \mathrm{PID}_i$, $\mathrm{HID}_j = h(\mathrm{FID}_j \| K_{\mathrm{CS}})$, $N_2 = h(A_1 \| K_{\mathrm{FN}} \|$

Table 1: The summary of authentication protocols.

Protocols	Cryptographic techniques	Limitations
Li et al. [19]	(1) Utilized digital signature (2) Utilized asymmetric encryption (3) Based on anonymous authentication	(1) Does not resist key compromise impersonation attacks (2) Does not provide user anonymity
Ying and Nayak [23]	(1) Utilized one-way hash function (2) Based on Diffie–Hellman problem (3) Based on anonymous authentication	(1) Does not resist replay attacks (2) Does not resist offline identity guessing attacks
Mohit et al. [24]	(1) Utilized one-way hash function (2) Based on smart card (3) Two-factor	(1) Does not resist impersonation attacks (2) Does not provide anonymity and untraceability (3) Does not provide mutual authentication
Yu et al. [25]	(1) Utilized one-way hash function (2) Based on smart card (3) Two-factor	(1) Does not resist sensor capture attacks (2) Does not resist impersonation attacks (3) Does not provide untraceability
Ma et al. [10]	(1) Utilized one-way hash function (2) Based on smart card(3) Utilized ECC	(1) Does not resist internal attacks (2) Does not resist stolen smart card attacks (3) Does not resist known session-specific temporary information attacks
Wazid et al. [34]	(1) Utilized one-way hash function (2) Based on anonymous authentication (3) Utilized ECC	—
Eftekhari et al. [9]	(1) Utilized one-way hash function (2) Based on anonymous authentication (3) Utilized ECC	—
Wu et al. [32]	(1) Utilized one-way hash function (2) Based on smart card (3) Utilized ECC (4) Two-factor	—

Table 2: Notations used in the proposed protocol.

Symbol	Description
V_i	The i-th vehicle
FN_j	The j-th fog node
CS	Cloud server
ID_i, FID_j, ID_{CS}	Identities of V_i, FN_j, and CS
PSW_i	Password of the V_i
K_{FN}	Shared key of FN_j and CS
K_{CS}	Secret key of CS
K_V	Counter value of V_i
SK	Session key

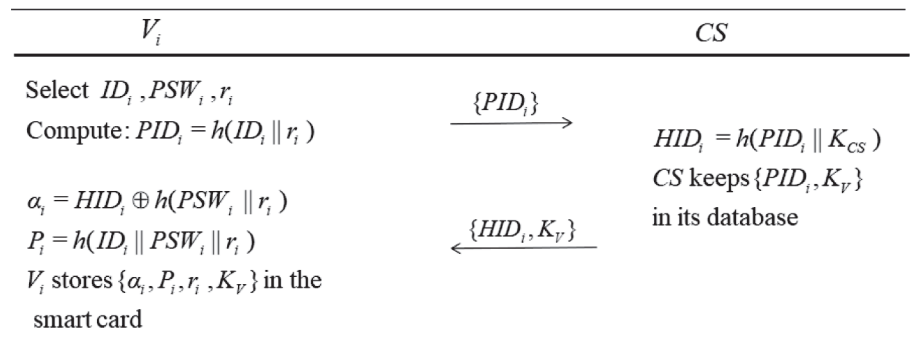

Figure 2: V_i registration phase.

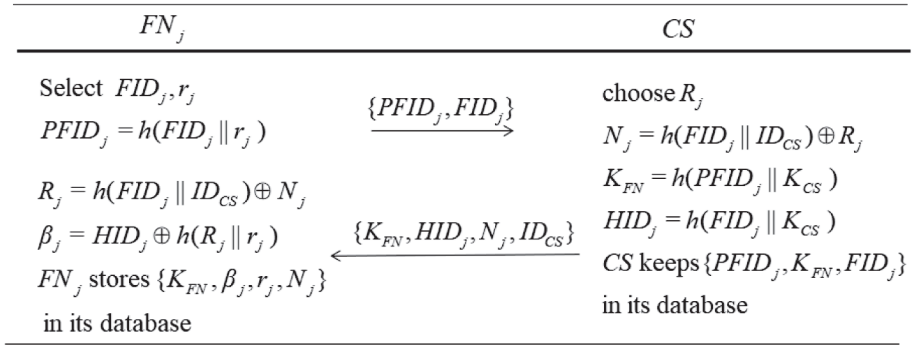

Figure 3: FN_j registration phase.

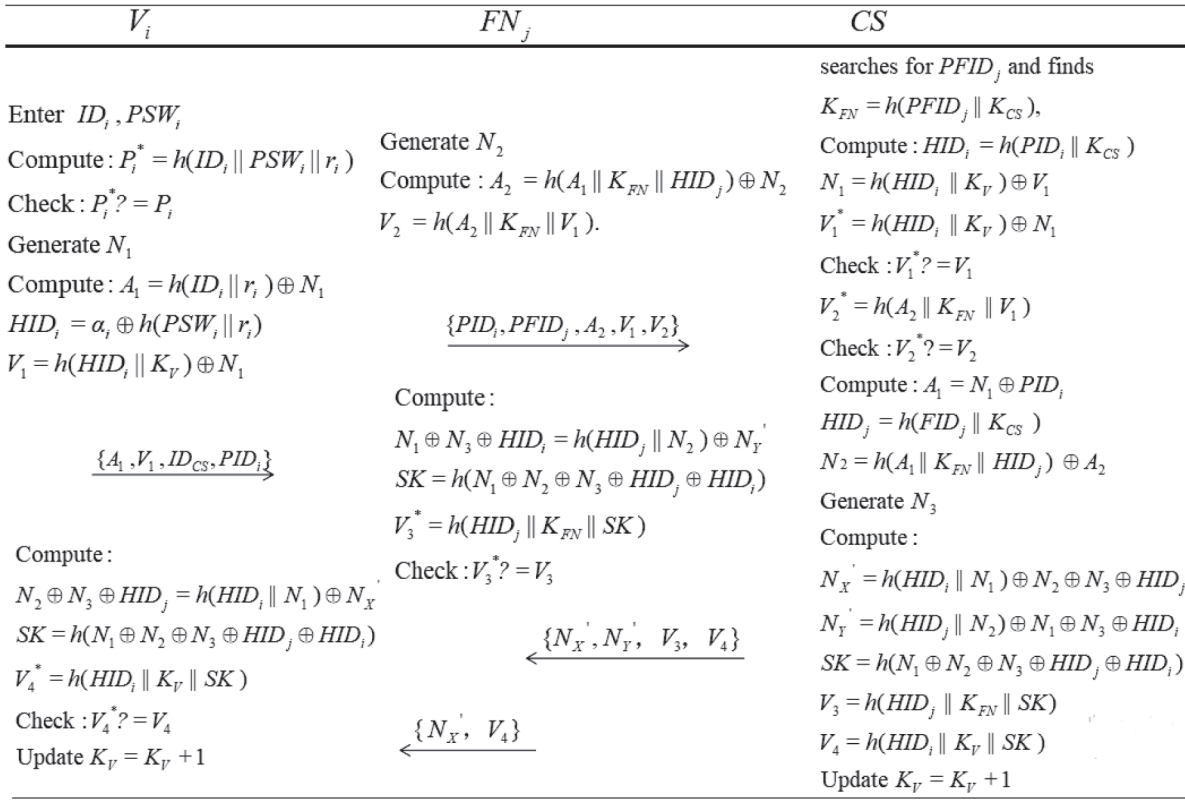

Figure 4: Login and authentication phase.

$\mathrm{HID}_j) \oplus A_2$, selects a random number N_3, and calculates $N_X' = h(\mathrm{HID}_i \| N_1) \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_j$, $N_Y' = h(\mathrm{HID}_j \| N_2) \oplus N_1 \oplus N_3 \oplus \mathrm{HID}_i$, $\mathrm{SK} = h(N_1 \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_j \oplus \mathrm{HID}_i)$, $V_3 = h(\mathrm{HID}_j \| K_{\mathrm{FN}} \| \mathrm{SK})$, $V_4 = h(\mathrm{HID}_i \| K_V \| \mathrm{SK})$. Then, it updates $K_V = K_V + 1$, and finally, CS sends message $M_3 = \{N_X', N_Y', V_3, V_4\}$ to FN_j.

(4) After receiving message M_3 from CS, FN_j calculates $N_1 \oplus N_3 \oplus \mathrm{HID}_i = h(\mathrm{HID}_j \| N_2) \oplus N_Y'$, $\mathrm{SK} = h(N_1 \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_j \oplus \mathrm{HID}_i)$, $V_3^* = h(\mathrm{HID}_j \| K_{\mathrm{FN}} \| \mathrm{SK})$, and compares $V_3^* \stackrel{?}{=} V_3$. If it is equal, it means that FN_j believes that CS is legal. Otherwise, the authentication process is terminated. Finally, FN_j sends message $M_4 = \{N_X', V_4\}$ to V_i.

(5) After receiving message M_4 from FN_j, V_i calculates $N_2 \oplus N_3 \oplus \mathrm{HID}_j = h(\mathrm{HID}_i \| N_1) \oplus N_X'$, $\mathrm{SK} = h(N_1 \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_j \oplus \mathrm{HID}_i)$, $V_4^* = h(\mathrm{HID}_i \| K_V \| \mathrm{SK})$, and compares $V_4^* \stackrel{?}{=} V_4$. If equal, it means that V_i believes that FN_j and CS are legal. Otherwise, the authentication process is terminated. Finally, V_i updates $K_V = K_V + 1$.

4. Security Analysis

4.1. BAN Logic. BAN logic is a formal security analysis method [35]. In this part, we use BAN logic to prove that vehicles, fog nodes, and cloud servers share a session key SK and further prove the correctness of our protocol. The rules used in BAN logic are shown in the references.

4.1.1. BAN Logic Rules

(1) Message-meaning rule: $(P| \equiv P \overset{K}{\leftrightarrow} Q,\ P \triangleleft \{X\}_K)/(P| \equiv Q| \sim X)$

(2) Freshness rule: $(P| \equiv \sharp(X))/(P| \equiv \sharp(X, Y))$

(3) Nonce-verification rule: $(P| \equiv \sharp(X),\ P| \equiv Q| \sim X)/(P| \equiv Q| \equiv X)$

(4) Jurisdiction rule: $(P| \equiv \sharp(X), P| \equiv Q| \sim X)/(P| \equiv Q| \equiv X)$

(5) Belief rule: $(P| \equiv X,\ P| \equiv Y)/(P| \equiv (X, Y))$

(6) Session key rule: $(P| \equiv \sharp(X),\ P| \equiv Q| \equiv X)/(P| \equiv P \overset{K}{\leftrightarrow} Q)$

4.1.2. Goals.

G1 $V_i| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j$

G2 $V_i| \equiv \mathrm{FN}_j| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j$

G3 $\mathrm{FN}_j| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j$

G4 $\mathrm{FN}_j| \equiv V_i| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j$

G5 $\mathrm{CS}| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j$

G6 $\mathrm{CS}| \equiv V_i| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j$

G7 $\mathrm{CS}| \equiv \mathrm{FN}_j| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j$

4.1.3. Idealizing Communication.

M1 $V_i \longrightarrow \mathrm{CS}: \{\mathrm{PID}_i, A_1, V_1\}$

M2 $\mathrm{FN}_j \longrightarrow \mathrm{CS}: \{\mathrm{PFID}_j, A_2, V_2\}$

M3 $\mathrm{CS} \longrightarrow \mathrm{FN}_j: \{N'_Y, V_3\}$

M4 $\mathrm{CS} \longrightarrow V_i: \{N'_X, V_4\}$

4.1.4. Initial State Assumptions

A1 $V_i| \equiv \sharp(N_1)$

A2 $\mathrm{FN}_j| \equiv \sharp(N_2)$

A3 $\mathrm{CS}| \equiv \sharp(N_3)$

A4 $\mathrm{CS}| \equiv V_i \overset{h(\mathrm{HID}_i \| K_V)}{\rightleftharpoons} \mathrm{CS}$

A5 $\mathrm{CS}| \equiv \sharp(N_1)$

A6 $\mathrm{CS}| \equiv V_i| \Longrightarrow N_1$

A7 $\mathrm{CS}| \equiv \mathrm{FN}_j \overset{h(\mathrm{A}_1 \| K_{\mathrm{FN}} \| \mathrm{HID}_j)}{\rightleftharpoons} \mathrm{CS}$

A8 $\mathrm{CS}| \equiv \sharp(N_2)$

A9 $\mathrm{CS}| \equiv \mathrm{FN}_j| \Longrightarrow N_2$

A10 $\mathrm{CS}| \equiv \mathrm{HID}_i$

A11 $\mathrm{CS}| \equiv \mathrm{HID}_{j h(\mathrm{HID}_j \| N_2)}$

A12 $\mathrm{FN}_j| \equiv \mathrm{FN}_j \rightleftharpoons \mathrm{CS}$

A13 $\mathrm{FN}_j| \equiv \mathrm{CS}| \Longrightarrow N_3$

A14 $\mathrm{FN}_j| \equiv \mathrm{CS}| \Longrightarrow \mathrm{HID}_j$

A15 $\mathrm{FN}_j| \equiv \sharp(N_1)$

A16 $\mathrm{FN}_j| \equiv \sharp(N_3)$

A17 $\mathrm{FN}_j| \equiv \sharp(\mathrm{HID}_j)$

A18 $V_i| \equiv V_i \overset{h(\mathrm{HID}_i \| N_1)}{\rightleftharpoons} \mathrm{CS}$

A19 $V_i| \equiv \mathrm{CS}| \Longrightarrow N_3$

A20 $V_i| \equiv \mathrm{CS}| \Longrightarrow \mathrm{HID}_j$

A21 $V_i| \equiv \sharp(N_2)$

A22 $V_i| \equiv \sharp(N_3)$

A23 $V_i| \equiv \sharp(\mathrm{HID}_j)$

4.1.5. Detailed Steps. By considering the message $M1$ and using the seeing rule, we get

$$\mathbf{S}1\text{: } \mathrm{CS} \triangleleft \left\{V_1\text{: } \langle N_1 \rangle_{h(\mathrm{HID}_i \| K_V)}, \mathrm{PID}_i, A_1\right\}. \tag{1}$$

Using $S1$, we get

$$\mathbf{S}2\text{: } \mathrm{CS} \triangleleft \left\{\langle N_1 \rangle_{h(\mathrm{HID}_i \| K_V)} \right. \tag{2}$$

Under the premise of assuming $A4$, using $S2$, and the message-meaning rule, we get

$$\mathbf{S}3\text{: } \mathrm{CS}\left| \equiv V_i \right| \sim N_1. \tag{3}$$

In the case of conclusion $S3$, using assumption $A5$, the freshness rule, and the nonce-verification (N-V) rule, we get

$$\mathbf{S}4\text{: } \mathrm{CS}\left| \equiv V_i \right| \equiv N_1. \tag{4}$$

In the case of conclusion $S4$, using assumption $A6$, and the jurisdiction rule, we get

$$\mathbf{S}5\text{: } \mathrm{CS}| \equiv N_1. \tag{5}$$

In addition, considering the message $M2$, we get

$$\mathbf{S}6\text{: } \mathrm{CS} \triangleleft \left\{\mathrm{PFID}_i, A_2\text{: } \langle N_2 \rangle_{h(A_1 \| K_{FN} \| \mathrm{HID}_j)}, V_2\right\}. \tag{6}$$

Using $S6$, we get

$$\mathbf{S}7\text{: } \mathrm{CS} \triangleleft \left\{\langle N_2 \rangle_{h(A_1 \| K_{FN} \| \mathrm{HID}_j)} \right. \tag{7}$$

Under the premise of assuming $A7$, using $S7$, and the message-meaning rule, we get

$$\mathbf{S}8\text{: } \mathrm{CS}\left| \equiv \mathrm{FN}_j \right| \sim N_2. \tag{8}$$

In the case of conclusion $S8$, using assumption $A8$, the freshness rule, and the nonce-verification (N-V) rule, we get

$$\mathbf{S}9\text{: } \mathrm{CS}\left| \equiv \mathrm{FN}_j \right| \equiv N_2. \tag{9}$$

In the case of conclusion $S9$, using assumption $A9$, and the jurisdiction rule, we get

$$\mathbf{S}10\text{: } \mathrm{CS}| \equiv N_2. \tag{10}$$

Because $\mathrm{SK} = h(N_1 \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_i \oplus \mathrm{HID}_j)$, according to the conclusions $A10$, $A11$, $S10$, and $S5$ and the belief rule, we get

$$\mathbf{S}11\text{: } \mathrm{CS}| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j, \quad (\mathbf{G}5). \tag{11}$$

Using $A5$, $S11$, and the SK rule, we get

$$\mathbf{S}12\text{: } \mathrm{CS}\left| \equiv V_i \right| \equiv V_i \overset{SK}{\leftrightarrow} \mathrm{FN}_j, \quad (\mathbf{G}6). \tag{12}$$

Using $A8$, $S11$, and the SK rule, we get

$$\mathbf{S}13\text{: } \mathrm{CS}\left| \equiv \mathrm{FN}_j \right| \equiv V_i \overset{\mathrm{SK}}{\leftrightarrow} \mathrm{FN}_j, \quad (\mathbf{G}7). \tag{13}$$

By considering the message $M3$ and using the seeing rule, we get

$$\mathbf{S}14\text{: } \mathrm{FN}_j \lhd \left\{ V_Y' \text{: } \langle N_1, N_3, \mathrm{HID}_i \rangle_{h\left(\mathrm{HID}_j \| N_2\right)}, V_3 \right\}. \tag{14}$$

Using $S14$, we get

$$\mathbf{S}15\text{: } \mathrm{FN}_j \lhd \left\{ \langle N_1, N_3, HID_i \rangle_{h\left(HID_j \| N_2\right)} \right. \tag{15}$$

Under the premise of assuming $A12$, using $S15$, and the message-meaning rule, we get

$$\mathbf{S}16\text{: } \mathrm{FN}_j | \equiv \mathrm{CS} | \sim \left(N_1, N_3, HID_i\right). \tag{16}$$

In the case of conclusion $S16$, using assumptions $A13$ and $A14$, the freshness rule, and the nonce-verification (N-V) rule, we get

$$\mathbf{S}17\text{: } \mathrm{FN}_j | \equiv \mathrm{CS} | \equiv \left(N_1, N_3, HID_i\right). \tag{17}$$

Applying this for each component, we get

$$\begin{aligned} &\mathbf{S}18\text{: } \mathrm{FN}_j | \equiv \mathrm{CS} | \equiv N_1, \\ &\mathbf{S}19\text{: } \mathrm{FN}_j | \equiv \mathrm{CS} | \equiv N_3, \\ &\mathbf{S}20\text{: } \mathrm{FN}_j | \equiv \mathrm{CS} | \equiv \mathrm{HID}_i. \end{aligned} \tag{18}$$

In the case of conclusion $S18$, using assumption $A15$, and the jurisdiction rule, we get

$$\mathbf{S}21\text{: } \mathrm{FN}_j | \equiv N_1. \tag{19}$$

In the case of conclusion $S22$, using assumption $A16$, and the jurisdiction rule, we get

$$\mathbf{S}22\text{: } \mathrm{FN}_j | \equiv N_1. \tag{20}$$

In the case of conclusion $S23$, using assumption $A17$, and the jurisdiction rule, we get

$$\mathbf{S}23\text{: } \mathrm{FN}_j | \equiv \mathrm{HID}_i. \tag{21}$$

Because $\mathrm{SK} = h(N_1 \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_i \oplus \mathrm{HID}_j)$, according to the conclusions $S21$, $S22$, and $S23$ and the belief rule, we get

$$\mathbf{S}24\text{: } \mathrm{FN}_j | \equiv V_i \overset{\mathrm{SK}}{\leftrightarrow} \mathrm{FN}_j, \quad (\mathbf{G}3). \tag{22}$$

Using $A15$, $S24$, and the SK rule, we get

$$\mathbf{S}25\text{: } \mathrm{FN}_j \left| \equiv V_i \right| \equiv V_i \overset{\mathrm{SK}}{\leftrightarrow} \mathrm{FN}_j, \quad (\mathbf{G}4). \tag{23}$$

By considering the message $M4$ and using the seeing rule, we get

$$\mathbf{S}26\text{: } V_i \lhd \left\{ V_X' \text{: } \langle N_2, N_3, \mathrm{HID}_j \rangle_{h\left(\mathrm{HID}_i \| N_1\right)}, V_4 \right\}. \tag{24}$$

Using $S26$, we get

$$\mathbf{S}27\text{: } V_i \lhd \left\{ \langle N_2, N_3, \mathrm{HID}_j \rangle_{h\left(\mathrm{HID}_i \| N_1\right)} \right. \tag{25}$$

Under the premise of assuming $A18$, using $S27$, and the message-meaning rule, we get

$$\mathbf{S}28\text{: } V_i | \equiv \mathrm{CS} | \sim \left(N_2, N_3, \mathrm{HID}_j\right). \tag{26}$$

In the case of conclusion $S28$, using assumption $A19$ and $A20$, the freshness rule, and the nonce-verification (N-V) rule, we get

$$\mathbf{S}29\text{: } V_i | \equiv \mathrm{CS} | \equiv \left(N_2, N_3, \mathrm{HID}_j\right). \tag{27}$$

Applying this for each component, we get

$$\begin{aligned} &\mathbf{S}30\text{: } V_i | \equiv \mathrm{CS} | \equiv N_2, \\ &\mathbf{S}31\text{: } V_i | \equiv \mathrm{CS} | \equiv N_3, \\ &\mathbf{S}32\text{: } V_i | \equiv \mathrm{CS} | \equiv \mathrm{HID}_j. \end{aligned} \tag{28}$$

In the case of conclusion $S30$, using assumptions $A21$, and the jurisdiction rule, we get

$$\mathbf{S}31\text{: } V_i | \equiv N_2. \tag{29}$$

In the case of conclusion $S31$, using assumptions $A22$, and the jurisdiction rule, we get

$$\mathbf{S}32\text{: } V_i | \equiv N_3. \tag{30}$$

In the case of conclusion $S32$, using assumptions $A23$, and the jurisdiction rule, we get

$$\mathbf{S}33\text{: } V_i | \equiv \mathrm{HID}_j. \tag{31}$$

Because $\mathrm{SK} = h(N_1 \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_i \oplus \mathrm{HID}_j)$, according to the conclusions $S31$, $S32$, and $S33$ and the belief rule, we get

$$\mathbf{S}34\text{: } V_i | \equiv V_i \overset{\mathrm{SK}}{\leftrightarrow} \mathrm{FN}_j, \quad (\mathbf{G}1). \tag{32}$$

Using $A21$, $S34$, and the SK rule, we get

$$\mathbf{S}35\text{: } V_i \left| \equiv \mathrm{FN}_j \right| \equiv V_i \overset{\mathrm{SK}}{\leftrightarrow} \mathrm{FN}_j, \quad (\mathbf{G}2). \tag{33}$$

4.2. Formal Security Analysis. In this part, we use the ROR model to formally prove the security of our proposed protocol. The ROR model judges the security of the protocol by calculating the session key SK probability of an ordinary situation [36, 37].

4.2.1. ROR Model. The protocol consists of three entities: vehicle, fog node, and cloud server. In the ROR model, we use $\Pi_{V_i}^x$, $\Pi_{\mathrm{FN}_j}^y$, and Π_{CS}^z to represent the x-th communication of the V_i, the y-th communication of the FN_j, and the z-th communication of the CS, respectively. We also define that the attacker A can have the following query capabilities, where $Z = \left\{ \Pi_{V_i}^x, \Pi_{\mathrm{FN}_j}^y, \Pi_{\mathrm{CS}}^z \right\}$.

Execute (Z): by performing this query operation, A can intercept the messages transmitted on the public channel.

Hash (string): by performing this query operation, A can obtain the hash value of the input string.

Send (Z, M): by performing this query operation, A can send message M to Z and receive the response from Z.

Corrupt (Z): by performing this query operation, A can obtain a party's secret values, such as some values in the smart card, long-term key, or temporary information.

Test (Z): by performing this query operation, A flips a coin c. If $c = 1$, A can obtain an accurate session key; if $c = 0$, A can obtain a random string of the same length as the session key.

4.2.2. Theorem. In the ROR model, assume A can perform execute, hash, send, corrupt, and test queries. Then, the probability that A can break the proposed protocol P in polynomial time is $\mathrm{Adv}_A^P(\xi) \le (q_{\mathrm{send}}/2^{l-1}) + (3q_{\mathrm{hash}}^2/2^l) + 2\max\{C' \cdot q_{\mathrm{send}}^{s'}, (q_{\mathrm{send}}/2^l)\} + ((q_{\mathrm{exe}} + q_{\mathrm{send}})^2/p)$, where q_{hash} represents the number of times hash queries are executed, q_{send} represents the number of times send queries are executed, q_{exe} represents the number of times execute queries are executed, l represents the bits of biological information, and C' and s are constants in Zipf's law.

4.2.3. Proof. We played five rounds of games, which were expressed as follows: GM_0 to GM_6. $\mathrm{Succ}_A^{\mathrm{GM}_i}(\xi)$ represents the event that A can win in the game GM_i. $\mathrm{Succ}_P^{A,\mathrm{GM}_i} = \Pr[\mathrm{Succ}_A^{\mathrm{GM}_i}]$ represents the advantage of A for winning GM_i. $\Pr[Z]$ is the probability of event Z. Adv_P^A represents the advantage A has in breaking the security of SK for protocol P. The specific steps of GM_i are as follows:

GM_0: GM_0 is the first-round game in the ROR model and a real attack. We choose a coin c to start the round. Therefore, in GM_0, we can obtain the probability that A can successfully break P as

$$\mathrm{Adv}_A^P = \left|2\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_0}\right] - 1\right|. \tag{34}$$

GM_1: GM_1 adds an execute query to GM_0. In GM_1, A can only obtain the messages transmitted on the public channel. After GM_1, A will query the session key SK through the test, but A cannot obtain five values $\{N_1, N_2, N_3, \mathrm{HID}_i, \mathrm{HID}_j\}$, so the probability that GM_0 is equal to that of GM_1 is

$$\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_1}\right] = \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_0}\right]. \tag{35}$$

GM_2: GM_2 adds a send query to GM_1. According to Zipf's law [38], we obtain

$$\left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_2}\right] - \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_1}\right]\right| \le \left(q_{\mathrm{send}}/2^l\right). \tag{36}$$

GM_3: GM_3 adds the hash query to GM_2. The maximum probability of text collision in transmission is $(q_{\mathrm{exe}} + q_{\mathrm{send}})^2/2p$, and we can obtain

$$\left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_3}\right] - \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_2}\right]\right| \le \frac{(q_{\mathrm{exe}} + q_{\mathrm{send}})^2}{2p} + \frac{q_{\mathrm{hash}}^2}{2^{l+1}}. \tag{37}$$

GM_4: in this round, we verify the security of the session key SK using two events. One is to obtain the long-term key of Π_{CS}^z to verify the perfect forward security, and the other is to obtain temporary information to verify that the protocol can resist the known session-specific temporary information attacks.

(1) Perfect forward security: using Π_{CS}^z, A attempts to obtain the private key K_{CS} of CS, or A uses $\Pi_{V_i}^x$ or $\Pi_{\mathrm{FN}_j}^y$ to obtain some secret values in the registration phase.

(2) Known session-specific temporary information attacks: A uses one of $\Pi_{V_i}^x$ or $\Pi_{\mathrm{FN}_j}^y$ or Π_{CS}^z to attempt to obtain temporary information.

For the first event, if A obtains the private key K_{CS} of CS, or the secret value of $\Pi_{V_i}^x$ and $\Pi_{\mathrm{FN}_j}^y$ in the registration phase, but A cannot get the random number $N_1, N_2, N_3, \mathrm{HID}_j$, it cannot calculate session key SK, where $\mathrm{SK} = h(N_1 \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_i \oplus \mathrm{HID}_j)$. For the second event, if A can obtain N_1, but the values of N_2 and N_3 are confidential, the SK cannot be calculated. Similarly, if N_2 and N_3 are leaked, SK cannot be calculated by A. Therefore, the probability of this round is

$$\left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_4}\right] - \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_3}\right]\right| \le \frac{q_{\mathrm{hash}}^2}{2^{l+1}}. \tag{38}$$

GM_5: in this round of the game, A uses the corrupt query to obtain the parameter $\{\alpha_i, P_i, r_i, K_V\}$ stored in the smart card, so A wants to conduct the offline key guessing attacks. V_i uses random numbers and passwords for registration, so A must guess $P_i = h(\mathrm{ID}_i \| \mathrm{PSW}_i \| r_i)$, but the probability of guessing a random number is $1/2^l$, which can be ignored. Using Zipf's law [38], we can obtain

$$\left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_5}\right] - \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_4}\right]\right| \le \max\left\{C' \cdot q_{\mathrm{send}}^{s'}, \frac{q_{\mathrm{send}}}{2^l}\right\}. \tag{39}$$

GM_6: this round of the game is to verify that protocol P can resist the impersonation attacks, A uses $h(N_1 \oplus N_2 \oplus N_3 \oplus \mathrm{HID}_i \oplus \mathrm{HID}_j)$ to query, and the game is terminated. Therefore, the probability that A can guess SK is

$$\left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_6}\right] - \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_5}\right]\right| \le \frac{q_{\mathrm{hash}}^2}{2^{l+1}}. \tag{40}$$

Because the probability of success and failure of the GM_6 is 1/2,

$$\begin{aligned}\frac{1}{2}\mathrm{Adv}_A^P &= \left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_0}\right] - \frac{1}{2}\right| \\ &= \left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_0}\right] - \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_6}\right]\right| \\ &= \left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_1}\right] - \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_6}\right]\right| \\ &\le \sum_{i=0}^{5}\left|\Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_{i+1}}\right] - \Pr\left[\mathrm{Succ}_A^{\mathrm{GM}_i}\right]\right| \\ &= \frac{q_{\mathrm{send}}}{2^l} + \frac{3q_{\mathrm{hash}}^2}{2^{l+1}} + \max \\ &\quad \cdot \left\{C' \cdot q_{\mathrm{send}}^{s'}, \frac{q_{\mathrm{send}}}{2^l}\right\} + \frac{\left(q_{\mathrm{hash}} + q_{\mathrm{send}}\right)^2}{2p}.\end{aligned} \tag{41}$$

Finally, we can obtain

$$\begin{aligned}\mathrm{Adv}_A^P \le \frac{q_{\mathrm{send}}}{2^{l-1}} + \frac{3q_{\mathrm{hash}}^2}{2^l} + 2\max\{C' \cdot \\ \cdot q_{\mathrm{send}}^{s'}, \frac{q_{\mathrm{send}}}{2^l}\} + \frac{\left(q_{\mathrm{exe}} + q_{\mathrm{send}}\right)^2}{p}.\end{aligned} \tag{42}$$

4.3. ProVerif. ProVerif is a formal automatic verification tool, which can verify confidentiality, identity, anonymity, and so on [39, 40]. In this paper, we use the ProVerif code to achieve vehicle registration, fog node registration, and authentication between the two parties and the CS and verify the security and effectiveness of our proposed protocol through ProVerif.

ProVerif demonstrates that the specific operation works as follows. Our protocol includes three entities: vehicle, fog node, and cloud server. The symbols and operation definitions used in ProVerif are shown in Figure 5.

The proof contains six events, as shown in Figure 6. The six events are veclestarted (), vecleauthored (), cloudserveracvehicle (), cloudserveracfognode (), fognodeaccloudserver (), and vecleaccloudserver (), indicating that the vehicle starts certification, the vehicle completes certification, the cloud server completes the vehicle certification, and the cloud server completes the fog node certification, respectively. The fog node completed the certification of the cloud server, and the vehicle completed the certification of the cloud server.

Then, we use ProVerif to query whether A can calculate the session key SK through the data transmitted on the common channel. The query operation is shown in Figure 7.

Finally, we get the verification result using the ProVerif tool, as shown in Figure 8. The result shows that A cannot calculate the session key SK of the V_i, FN_j, and CS.

4.4. Informal Security Analysis. This part is an informal security analysis of our proposed agreement. We have proved that the protocol can meet common security requirements. The specific proof is as follows.

4.4.1. Mutual Authentication. In the authentication phase, with the help of CS, mutual authentication between V_i and FN_j is realized. V_1 in message M_1 is the value CS uses to authenticate V_i, V_2 in message M_2 is the value CS uses to authenticate FN_j, and V_3 and V_4 in message M_3 are the values CS uses to authenticate FN_j and V_i, respectively. Therefore, the mutual authentication among V_i, FN_j, and CS is realized in the authentication phase.

4.4.2. Replay Attacks. In this protocol, we use cumulative value K_V to resist replay attacks. In the V_i registration phase, we initialize K_V to 0. As the session progresses, it carries out +1 operation on the value K_V, saves it to its database after CS authenticates V_i, FN_j, and carries out the necessary calculation. After CS authenticates V_i and generates the session key, it also carries out the +1 operation on the value K_V and saves it to the smart card. In this manner, K_V on both sides is synchronous and equal, and the session process is completed smoothly. If A repeatedly sends message M_1 intercepted in the public channel, CS continues to calculate the value K_V + 1 in the authentication phase. Value V_4 generated using K_V is not equal to value V_4 calculated by V_i using K_V stored in its smart card, because the value K_V in the smart card of V_i cannot keep up with the CS update speed, so the authentication fails. Thus, our protocol can resist replay attacks.

4.4.3. Man-in-the-Middle Attacks. Suppose that A can intercept the message $M_1 = \{A_1, V_1, \mathrm{ID}_{\mathrm{CS}}, \mathrm{PID}_i\}$ transmitted on the public channel between V_i and FN_j. Since A cannot obtain the information $\{\alpha_i, K_V, r_i\}$ in the smart card and the identity ID_i of V_i, A cannot calculate the values $\{K_V, \mathrm{HID}_i, N_1\}$ required for V_1, where $V_1 = h(\mathrm{HID}_i \| K_V) \oplus N_1$. Therefore, after A tampers with M_1, it cannot pass the authentication of FN_j. Similarly, because the privacy value is unknown, A cannot calculate the authentication value V_2, V_3, or V_4 and cannot complete the verification after intercepting the information M_2, M_3, or M_4. Therefore, our protocol can resist man-in-the-middle attacks.

4.4.4. User Anonymity. The real identities of V_i and FN_j are transmitted on the secure channel and are protected by pseudoidentity PID_i and PFID_j in the authentication phase. The anonymity of V_i and FN_j is ensured. Therefore, our protocol can provide user anonymity.

4.4.5. Untraceability. If A wants to trace the V_i, it intercepts the messages $\{M_1, M_2, M_3, M_4\}$ transmitted on the common channel. Since the random numbers $\{N_1, N_2, N_3\}$ are used, this means that messages $\{M_1, M_2, M_3, M_4\}$ are different during each session. In addition, A cannot obtain the random numbers $\{N_1, N_2, N_3\}$, so A cannot be traced back to V_i. Therefore, our protocol can provide untraceability.

5. Security and Performance Comparisons

In this part, we compare our protocol with those of Ma et al. [10], Wazid et al. [34], Eftekhari et al. [9], and Wu et al. [32] in terms of security, computational cost, and communication cost.

```
(*  channel*)
free ch :channel. (*  public channel *)
free sch: channel [private]. (*  secure channel, used for registering *)
(*  shared keys *)
free SKv : bitstring  [private].
free SKf : bitstring  [private].
free SKc : bitstring  [private].
(* constants  *)
free  Kcs:bitstring [private].
free  P:bitstring.
free  B:bitstring.
free  yj:bitstring.
(*  functions  &  reductions & equations  *)
fun h(bitstring) :bitstring. (*  hash function *)
fun mult(bitstring,bitstring) :bitstring. (* scalar multiplication operation *)
fun add(bitstring,bitstring):bitstring. (* Addition operation *)
fun sub(bitstring,bitstring):bitstring. (* Subtraction operation *)
fun mod(bitstring,bitstring):bitstring. (* modulus operation *)
fun con(bitstring,bitstring):bitstring. (* concatenation operation *)
reduc forall m:bitstring, n:bitstring; getmess(con(m,n))=m.
fun xor(bitstring,bitstring):bitstring. (*  XOR operation *)
equation forall m:bitstring, n:bitstring; xor(xor(m,n),n)=m.
fun Gen(bitstring):bitstring. (* Generator operation *)
fun Rep(bitstring,bitstring):bitstring.
```

Figure 5: The definition in the ProVerif tool.

5.1. Security Comparisons. When comparing protocol security, we use ✓ to indicate that the protocol can resist the attacks and × to indicate that the protocol cannot resist the attacks. The results of comparing protocol security are shown in Table 3. It can be seen that our protocol can resist known attacks and have better security. Ma et al.'s protocol [10] cannot provide user anonymity and untraceability and is vulnerable to impersonation attacks and known session-specific temporary information attacks. The protocols in [9, 32, 34] and our protocol are secure.

5.2. Performance Comparison. Performance analysis is conducted from the aspects of computational cost and communication cost. We analyze and compare the computational cost from the login authentication phase of each protocol. The computational cost of XOR and join operations is negligible. The computational cost comparison is shown in Table 4. It is obvious that the protocols of Ma et al. [10], Wazid et al. [34], Eftekhari et al. [9], and Wu et al. [32] perform point multiplication, Wazid et al. [34] and Wu et al. [32] perform fuzzy extraction, and Wazid et al.'s protocol [34] and Eftekhari et al. [9] also perform ECC point addition. Only our proposed protocol performs the hash operation, so its computational cost is less.

Here, T_{pm} represents the time taken to perform a point multiplication operation, T_{pa} represents the time taken to execute an ECC point addition, T_f represents the time taken to execute a fuzzy extraction function, and T_h represents the time taken to execute a hash operation.

In the comparison of communication cost, we assume that the length of the identity and the random number are 160 bits, the length of the timestamp is 32 bits, the length of the one-way hash function is 256 bits, and the length of ECC point is 320 bits. Therefore, based on our assumption, the communication costs of the protocols of Ma et al. [10], Wazid et al. [34], Eftekhari et al. [9], and Wu et al. [32] are 4512 bits, 3488 bits, 4416 bits, and 4448 bits. Here, we illustrate our

```
(*  ----------------Vehicle's process----------------  *)
let ProcessVehicle=new IDi : bitstring; (* the Vehicle's ID *)
                    new PSWi : bitstring;
                    new ri : bitstring;
                    let PIDi=h(con(IDi,ri)) in
                    out(sch,(PIDi));
                    in(sch,(xHIDi:bitstring,xKv:bitstring));
                    let ai=xor(xHIDi,h(con(IDi,ri))) in
                    let Pi=h(con(con(IDi,PSWi),ri)) in
                    !(event VehicleStarted();
                    let Pi'=h(con(con(IDi,PSWi),ri)) in
                    if Pi'=Pi   then
                    new N1:bitstring;
                    let A1=xor(h(con(IDi,ri)),N1) in
                    let HIDi=xor(ai,h(con(PSWi,ri))) in
                    let V1=xor(h(con(HIDi,xKv)),N1) in
                    new IDcs:bitstring;
                    out(ch,(A1,V1,IDcs,PIDi)); (*--------authentication--------- *)
                    event  VehicleAuthed();
                    in(ch,(xNx':bitstring,xV4:bitstring));
                    (*let xor(xor(N2,N3),HIDj)=xor(h(con(HIDi,N1)),xNx') in *)
                    let P=xor(h(con(HIDi,N1)),xNx') in
                    let SKv=h(xor(xor(xor(P,HIDi),N1),HIDi)) in
                    let V4'=h(con(con(HIDi,xKv),SKv)) in
                    if V4'=xV4    then  event VehicleAcCloudServer();0).
(* ------FNj process------- *)
let ProcessFNj=new FIDj:bitstring;
                    new rj:bitstring;
                    let PFIDj=h(con(FIDj,rj)) in
                    out(sch,(FIDj,PFIDj));
                    in(sch,(yKFN:bitstring,yHIDj:bitstring,yNj:bitstring,yIDcs:bitstring));
                    let Rj=xor(h(con(FIDj,yIDcs)),yNj) in
                    ! (in(ch,(yA1:bitstring,yV1:bitstring,yIDcs:bitstring,yPIDi:bitstring));
                    new N2:bitstring;
                    let A2=xor(h(con(con(yA1,yKFN),yHIDj)),N2) in
                    let V2=h(con(con(A2,yKFN),yV1)) in
                    out(ch,(yPIDi,PFIDj,A2,yV1,V2));
                    in(ch,(yNx':bitstring,yNY:bitstring,yV3:bitstring,yV4:bitstring));
                    let Q=xor(h(con(yHIDj,N2)),yNY) in
                    let SKf=h(xor(xor(Q,N2),yHIDj)) in
                    let V3'=h(con(con(yHIDj,yKFN),SKf)) in
                    if V3'=yV3    then event FogNodeACcloudServer();
                    out (ch,(yNx',yV4));0).
```

```
(* --------CloudServer's process--------- *)
let VehicleReg= in(sch,(zPIDi:bitstring));
                    let HIDi=h(con(zPIDi,Kcs)) in
                    new Kv:bitstring;
                    out(sch,(HIDi,Kv));
                    0.
let FogNodeReg=in(sch,(zPFIDj:bitstring,zFIDj:bitstring));
                    new Rj:bitstring;
                    new IDcs:bitstring;
                    let Nj=xor(h(con(zFIDj,IDcs)),Rj) in
                    let KFN=h(con(zPFIDj,Kcs)) in
                    let HIDj=h(con(zFIDj,Kcs)) in
                    out(sch,(KFN,HIDj,Nj,IDcs));
                    0.
let
CloudServerAuth=in(ch,(zPIDi:bitstring,zPFIDj:bitstring,zA2:bitstring,zV1:bitstring,zV2:bitstring));
                    let KFN=h(con(zPFIDj,Kcs)) in
                    let HIDi=h(con(zPIDi,Kcs)) in
                    new Kv:bitstring;
                    let N1=xor(h(con(HIDi,Kv)),zV1)in
                    let V1'=xor(h(con(HIDi,Kv)),N1) in
                    if V1'=zV1    then event CloudServerAcVehicle();
                    let V2'=h(con(con(zA2,KFN),zV1)) in
                    if V2'=zV2    then event CloudServerAcFogNode();
                    let A1=xor(N1,zPIDi) in
                    new FIDj:bitstring;
                    let HIDj=h(con(FIDj,Kcs)) in
                    let N2=xor(h(con(con(A1,KFN),HIDj)),zA2) in
                    new N3:bitstring;
                    let P=xor(xor(N2,N3),HIDj) in
                    let Q=xor(xor(N1,N3),HIDi) in
                    let Nx'=xor(h(con(HIDi,N1)),P) in
                    let NY=xor(h(con(HIDj,N2)),Q) in
                    let SKc=h(xor(xor(xor(xor(N1,N2),N3),HIDj),HIDi)) in
                    let V3=h(con(con(HIDj,KFN),SKc)) in
                    let V4=h(con(con(HIDi,Kv),SKc)) in
                    out(ch,(Nx',NY,V3,V4));
                    0.
let ProcessCloudServer= VehicleReg |  FogNodeReg |   CloudServerAuth.
(* --------- main---------- *)
process
                    (!ProcessVehicle  |  !ProcessFNj | !ProcessCloudServer  )
```

Figure 6: Process in the ProVerif tool.

```
(*  queries  *)
query attacker(SKv).
query attacker(SKf).
query attacker(SKc).
query inj-event(VehicleAuthed()) ==> inj-event(VehicleStarted()).
query inj-event(CloudServerAcFogNode()) ==> inj-
event(CloudServerAcVehicle()).
query inj-event(VehicleAcCloudServer()) ==> inj-
event(FogNodeACcloudServer()).
(* event *)
event VehicleStarted().
event VehicleAuthed().
event CloudServerAcVehicle().
event CloudServerAcFogNode().
event FogNodeACcloudServer().
event VehicleAcCloudServer().
```

Figure 7: Queries and events in the ProVerif tool.

protocol as an example to show the specific analysis. In our protocol, the messages transmitted in the login authentication phase are $M_1 = \{A_1, V_1, \mathrm{ID}_{\mathrm{CS}}, \mathrm{PID}_i\}$, $M_2 = \{A_2, V_1, V_2, \mathrm{PFID}_j, \mathrm{PID}_i\}$, $M_3 = \{N'_X, N'_Y, V_3, V_4\}$, and $M_4 = \{N'_X, V_4\}$, where $\{A_1, A_2, N'_X, N'_Y\}$ are random strings, $\mathrm{ID}_{\mathrm{CS}}$ is an identity, and $\{V_1, V_2, V_3, V_4, \mathrm{PFID}_j, \mathrm{PID}_i\}$ are hash values. Therefore, the total communication cost of our proposed protocol is 2336 bits. The comparison of communication cost is shown in Table 5. Obviously, the communication cost of our proposed protocol is less.

In the security comparison, we found that Ma et al.'s protocol [10] cannot provide user anonymity and untraceability and is vulnerable to impersonation attacks and known session-specific temporary information attacks.

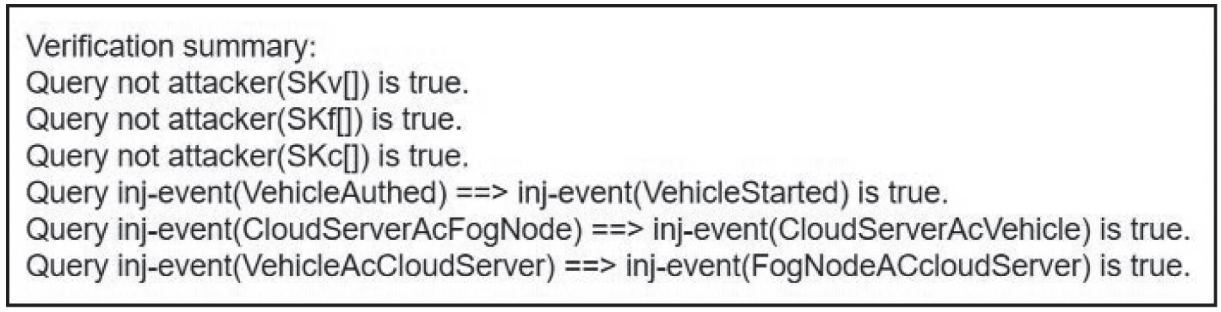

Figure 8: Results in the ProVerif tool.

Table 3: Comparisons of security.

Security properties	[10]	[34]	[9]	[32]	Ours
Mutual authentication	✓	✓	✓	✓	✓
User anonymity	×	✓	✓	✓	✓
Perfect forward secrecy	✓	—	✓	✓	✓
Man-in-the-middle attacks	✓	—	—	✓	✓
Impersonation attacks	✓	✓	✓	✓	✓
Known session-specific temporary information attacks	×	—	✓	✓	✓
Untraceability	×	✓	✓	✓	✓
Offline password guessing attacks	✓	✓	—	✓	✓
Replay attacks	✓	✓	✓	✓	✓

Table 4: Computational cost comparison.

Protocol	V_i	FN_j	CS	Total
Ma et al. [10]	$3T_{pm} + 4T_h$	$4T_{pm} + 4T_h$	$10T_{pm} + 11T_h$	$17T_{pm} + 19T_h$
Wazid et al. [34]	$3T_{pm} + 2T_f + 22T_h$	$2T_{pm} + T_{pa} + 14T_h$	$3T_h$	$5T_{pm} + 2T_{pa} + 2T_f + 43T_h$
Eftekhari et al. [9]	$3T_{pm} + T_{pa} + 11T_h$	$3T_{pm} + T_{pa} + 12T_h$	$3T_{pm} + 2T_{pa} + 15T_h$	$9T_{pm} + 4T_{pa} + 38T_h$
Wu et al. [32]	$2T_{pm} + T_f + 8T_h$	$4T_{pm} + 5T_h$	$4T_{pm} + 13T_h$	$10T_{pm} + T_f + 26T_h$
Ours	$7T_h$	$5T_h$	$11T_h$	$23T_h$

Table 5: Communication cost comparison.

Protocol	Round	Communication cost (bits)
Ma et al. [10]	4	4512
Wazid et al. [34]	3	3488
Eftekhari et al. [9]	4	4416
Wu et al. [32]	4	4448
Ours	4	2336

Although the protocols of [9, 32, 34] can resist known security attacks, the overhead in the aspect of computational cost and communication cost is much more than that of our proposed protocol. Therefore, our protocol is better in terms of security and performance.

6. Conclusions

In this paper, we first review the AKE protocol in IoV and SIoV, and then, we propose a lightweight and authenticated key agreement protocol using fog nodes. The security analysis of the protocol is conducted by using BAN, ROR, and ProVerif. The comparison of security and performance shows that the protocol achieves higher performance in terms of computing power and communication cost compared with other protocols. In future research, we will focus on improving the security and performance of the protocol in SIoV.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This article was supported by the Guangxi Key Laboratory of Trusted Software (no. KX202033).

References

[1] S. Arasteh, S. F. Aghili, and M. Hamid, "A new lightweight authentication and key agreement protocol for internet of things," in *Proceedings of the 2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, pp. 52–59, IEEE, Tehran, Iran, September 2016.

[2] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for internet of things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[3] X. Hu, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy-preserving authentication protocol for heterogeneous systems in iiot," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11713–11724, 2020.

[4] K. Park, Y. Park, A. K. Das, S. Yu, J. Lee, and Y. Park, "A dynamic privacy-preserving key management protocol for v2g in social internet of things," *IEEE Access*, vol. 7, pp. 76812–76832, 2019.

[5] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[6] S. Ahmed, S. Kumari, M. A. Saleem, K. Agarwal, K. Mahmood, and M.-H. Yang, "Anonymous key-agreement protocol for v2g environment within social internet of vehicles," *IEEE Access*, vol. 8, pp. 119829–119839, 2020.

[7] T. A. Butt, R. Iqbal, K. Salah, M. Aloqaily, and Y. Jararweh, "Privacy management in social internet of vehicles: review, challenges and blockchain based solutions," *IEEE Access*, vol. 7, pp. 79694–79713, 2019.

[8] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu, and X. Wu, "Energy-aware dynamic resource allocation in uav assisted mobile edge computing over social internet of vehicles," *IEEE Access*, vol. 6, pp. 56700–56715, 2018.

[9] S. A. Eftekhari, M. Nikooghadam, and M. Rafighi, "Security-enhanced three-party pairwise secret key agreement protocol for fog-based vehicular ad-hoc communications," *Vehicular Communications*, vol. 28, Article ID 100306, 2021.

[10] M. Ma, D. He, H. Wang, N. Kumar, and K.-K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8065–8075, 2019.

[11] I. Azimi, A. Anzanpour, A. M. Rahmani, P. Liljeberg, and T. Salakoski, "Medical warning system based on internet of things using fog computing," in *Proceedings of the 2016 international workshop on big data and information security (IWBIS)*, pp. 19–24, IEEE, Jakarta, Indonesia, October 2016.

[12] B. Ismail, A. Sari, and P. Österberg, "Security implications of fog computing on the internet of things," in *Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, IEEE, Las Vegas, NV, USA, January 2019.

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

[14] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE wireless communications*, vol. 13, no. 5, pp. 8–15, 2006.

[15] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: efficient conditional privacy preservation protocol for secure vehicular communications," in *Proceedings of the IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pp. 1229–1237, IEEE, Phoenix, AZ, USA, April 2008.

[16] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proceedings of the IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pp. 246–250, IEEE, Phoenix, AZ, USA, April 2008.

[17] J. Cui, L. Wei, H. Zhong, J. Zhang, Y. Xu, and L. Liu, "Edge computing in vanets-an efficient and privacy-preserving cooperative downloading scheme," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1191–1204, 2020.

[18] X. Hu, J. Chen, M. Qian, and Y. Zhao, "Conditional privacy-preserving authentication protocol with dynamic membership updating for vanets," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[19] J. Li, H. Lu, and M. Guizani, "Acpn: a novel authentication framework with conditional privacy-preservation and non-repudiation for vanets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938–948, 2014.

[20] D. Amit, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4359–4373, 2017.

[21] M.-C. Chuang and J.-F. Lee, "Team: trust-extended authentication mechanism for vehicular ad hoc networks," *IEEE Systems Journal*, vol. 8, no. 3, pp. 749–758, 2013.

[22] S. Kumari, M. Karuppiah, X. Li, F. Wu, A. K. Das, and V. Odelu, "An enhanced and secure trust-extended authentication mechanism for vehicular ad-hoc networks," *Security and Communication Networks*, vol. 9, no. 17, pp. 4255–4271, 2016.

[23] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10626–10636, 2017.

[24] P. Mohit, R. Amin, and G. P. Biswas, "Design of authentication protocol for wireless sensor network-based smart vehicular system," *Vehicular Communications*, vol. 9, pp. 64–71, 2017.

[25] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, "Secure authentication protocol for wireless sensor networks in vehicular communications," *Sensors*, vol. 18, no. 10, p. 3191, 2018.

[26] M. J. Sadri and M. Rajabzadeh Asaar, "A lightweight anonymous two-factor authentication protocol for wireless sensor networks in internet of vehicles," *International Journal of Communication Systems*, vol. 33, no. 14, Article ID e4511, 2020.

[27] T.-Y. Wu, Z. Lee, L. Yang, and C.-M. Chen, "A provably secure authentication and key exchange protocol in vehicular ad hoc networks," *Security and Communication Networks*, vol. 2021, Article ID 9944460, 17 pages, 2021.

[28] S. Bitam, A. Mellouk, and S. Zeadally, "Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 96–102, 2015.

[29] Q. Jiang, J. Ni, J. Ma, L. Yang, and X. Shen, "Integrated authentication and key agreement framework for vehicular cloud computing," *IEEE Network*, vol. 32, no. 3, pp. 28–35, 2018.

[30] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, and S. Cherkaoui, "RSU cloud and its resource management in support of enhanced vehicular applications," in *Proceedings of the 2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 127–132, IEEE, Austin, TX, USA, December 2014.

[31] Y. Rong, X. Huang, J. Kang et al., "Cooperative resource management in cloud-enabled vehicular networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7938–7951, 2015.

[32] T.-Y. Wu, Z. Lee, L. Yang, J.-N. Luo, and R. Tso, "Provably secure authentication key exchange scheme using fog nodes in vehicular ad hoc networks," *The Journal of Supercomputing*, vol. 77, no. 7, pp. 6992–7020, 2021.

[33] A. Maria, V. Pandi, J. Deborah Lazarus, M. Karuppiah, and M. S. Christo, "BBAAS: blockchain-based anonymous authentication scheme for providing secure communication in

vanets," *Security and Communication Networks*, vol. 2021, Article ID 6679882, 11 pages, 2021.

[34] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. Park, "AKM-IOV: authenticated key management protocol in fog computing-based internet of vehicles deployment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8804–8817, 2019.

[35] R. M. Needham, M. Burrows, and M. Abadi, "A logic of authentication," *Proceedings of the Royal Society of London. Mathematical and physical sciences Series*, vol. 426, no. 1871, pp. 233–271, 1989.

[36] C. Ran, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *Journal of the ACM*, vol. 51, no. 4, pp. 557–594, 2004.

[37] T.-Y. Wu, L. Yang, Z. Lee, C.-M. Chen, J.-S. Pan, and S. K. Hafizul Islam, "Improved ECC-based three-factor multiserver authentication scheme," *Security and Communication Networks*, vol. 2021, Article ID 6627956, 14 pages, 2021.

[38] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[39] B. Blanchet, V. Cheval, X. Allamigeon, and B. Smyth, "Proverif: cryptographic protocol verifier in the formal model; 2012," 2019, https://prosecco.gforge.inria.fr/personal/bblanche/proverif/.

[40] Y. Luo, W. M. Zheng, and Y.-C. Chen, "An anonymous authentication and key exchange protocol in smart grid," *Journal of Network Intelligence*, vol. 6, no. 2, pp. 206–215, 2021.