Data Security of Fog and Edge Computing-Based Social Internet of Vehicles - Research Article

International Journal of
Distributed
Sensor Networks

International Journal of Distributed Sensor Networks
2022, Vol. 18(6)

DOI: 10.1177/15501329221104332
journals.sagepub.com/home/dsn
SAGE

A provably secure and lightweight mutual authentication protocol in fog-enabled social Internet of vehicles

Zhen Li[1], Qingkai Miao[1], Shehzad Ashraf Chaudhry[2] and Chien-Ming Chen[1]

Abstract

The Internet of vehicles technology has developed rapidly in recent years and has become increasingly important. The social Internet of vehicles provides better resources and services for the development of the Internet of vehicles and provides better experience for users. However, there are still many security problems in social vehicle networking environments. Once the vehicle is networked, the biggest problem is data security according to the three levels of data collection, intelligent analysis, and decision control of the Internet of vehicles. Recently, Wu et al. proposed a lightweight vehicle social network security authentication protocol based on fog nodes. They claimed that their security authentication protocol could resist various attacks. However, we found that their authentication protocols are vulnerable to internal attacks, smart card theft attacks, and lack perfect forward security. In this study, we propose a new protocol to overcome these limitations. Finally, security and performance analyses show that our protocol perfectly overcomes these limitations and exhibits excellent performance and efficiency.

Keywords

Fog node, authentication, social Internet of vehicles

Date received: 21 February 2022; accepted: 4 May 2022

Handling Editor: Yanjiao Chen

Introduction

At the Information Society Summit held in 2005, the International Telecommunication Union (ITU) formally introduced the concept of the Internet of Things (IoT) in the form of an Internet report. IoT is based on the Internet, which uses radio frequency automatic identification, wireless data communication, and other technologies, to achieve automatic identification of objects and information interconnection and sharing, to build a "Internet of things" that encompasses everything in the world. The scope of application of the IoT is gradually expanding, and its application in various industries, agriculture, transportation, and others has promoted the development of intelligence in these fields, making resources allocation more rational and improving the efficiency of these industries. The application in life-related areas, such as smart warehouses, smart medical care, smart electricity, and tourism services has substantially improved the quality of people's lives, from the scope of services and the way they are provided to the quality of services. Fog computing is an extension of cloud computing, an IoT-based distributed computing infrastructure that can use devices in edge networks to enable the delivery of data with extremely low latency. The application of fog computing reduces

---

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China
[2]Istanbul Gelisim University, Istanbul, Turkey

Corresponding author:
Chien-Ming Chen, College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China.
Email: chienmingchen@ieee.org

inter-network distances, increases efficiency, and reduces the amount of data required to be transmitted to the cloud for processing, analysis, and storage. Fog nodes are a key component of the fog computing architecture, and they can appear in different forms and be deployed in a variety of environments.

The Internet of Vehicles (IoV) is an automotive mobile IoT technology that provides different functional services in the operation of vehicles through advanced sensor technology, communication technology, data processing technology, network technology, and information dissemination technology. The devices on the vehicles effectively use the information in the network platform. The IoV can provide spacing between vehicles and reduce the risk of collisions; it can help vehicle owners navigate in real-time and improve the efficiency of traffic operations by communicating with other vehicles and network systems. With the advances in the application of IoT, IoT technology is being combined with social networks to form a new network called the social Internet of things (SIoT). The IoT will include not only the association of things and things and people, but also introduces the relationship between people and people, thus better depicting the connected world of all things. SIoT is a new application of IoT technology in social networks. Ordinary objects in our lives can be informatized in real-time using IoT's sensing and monitoring technology, and the information of the objects can be displayed online through network technology, cloud computing technology, and cloud storage technology.

With the development of modern technology, the IoV also requires the organic combination of traditional IoV functions and social networking of vehicles, resulting in the rise of social networking of vehicles (SIoV). The SIoV is a social approach to increase the viscosity of the user, thereby maintaining the profitability of SIoV and the related information reserve.

SIoV provides better resources and services for the development of IoV. Telematics can be better implemented and telematics services can be better enhanced, only by continuously improving the functions of social telematics enhancing the popularity of telematics. In the SIoV environment, relevant information is entered into the telematics database in the background, and then the vehicle owner can use the telematics social services like a social software, which can always keep learning to obtain information and help the vehicle owner to improve the efficiency of the trip, and even enable the vehicle's remote pre-diagnosis of itself to improve safety. The typical structure of SIoV is shown in Figure 1, which mainly includes vehicle, roadside unit (RSU), a fog node, and a cloud server (*CS*). The cloud server is an infrastructure as a service (IaaS) service that integrates computing, storage, and network resources based on a WEB service that provides an elastic cloud technology with customizable cloud hosting configurations. Vehicles are tangible users and beneficiaries. Vehicles can communicate with each other and owners can access information, share location, and so on, to make travel safer and smarter. The RSU can collect information about nearby vehicles, send it to the fog node, and receive information from the fog node. In the telematics environment, the deployment of fog nodes is strongly influenced by geographical location; however, the prevalence of content within the coverage area varies greatly, because fog nodes are usually deployed in different areas and the cached content has certain geographical characteristics. A fog node is responsible for collecting and processing data of vehicles in a certain area, and subsequently, it transmits the collected data to the cloud server, reducing the computational load on the cloud server.

However, there are still many security issues in the SIoV environment. Once a car is connected, the biggest problem is data security according to the three levels of data collection, intelligent analysis, and decision control. If we want to achieve data interoperability and data sharing, particularly, if we want to achieve decision control, ensuring data security is the most challenging issue of entire vehicle networking. In addition to traditional solution techniques such as authentication and access control, two other typical issues are how to verify the reliability of the data and protect the privacy of the data. This reflects the importance and criticality of data encryption, which encrypts data to achieve data concealment and thus protect data security. Encryption requires negotiation of a common session key between the participating actors to achieve reliable transmission. Wu et al. proposed a fog node–based secure authentication protocol for vehicular social networks and an authentication protocol that ensures user anonymity and security. Wu et al. claimed that their proposed secure authentication protocol was resistant to various attacks. However, we find that their authentication protocol is vulnerable to offline password guessing attacks, smart card theft attacks, and lacks perfect forward security, and there are also some design issues in this scheme. Here, we present these issues and make recommendations.

The main contributions of this article are as follows:

1. We perform a security analysis of the authentication protocol proposed by Wu et al. for SIoV and find that their authentication protocol is vulnerable to insider attacks, smart card theft attacks, and lacks perfect forward security. As a user, we focus on protecting the data anonymity and security of the vehicle, prioritize data security in the protocol design, and propose a new scheme to improve the shortcomings of Wu et al.'s protocol.

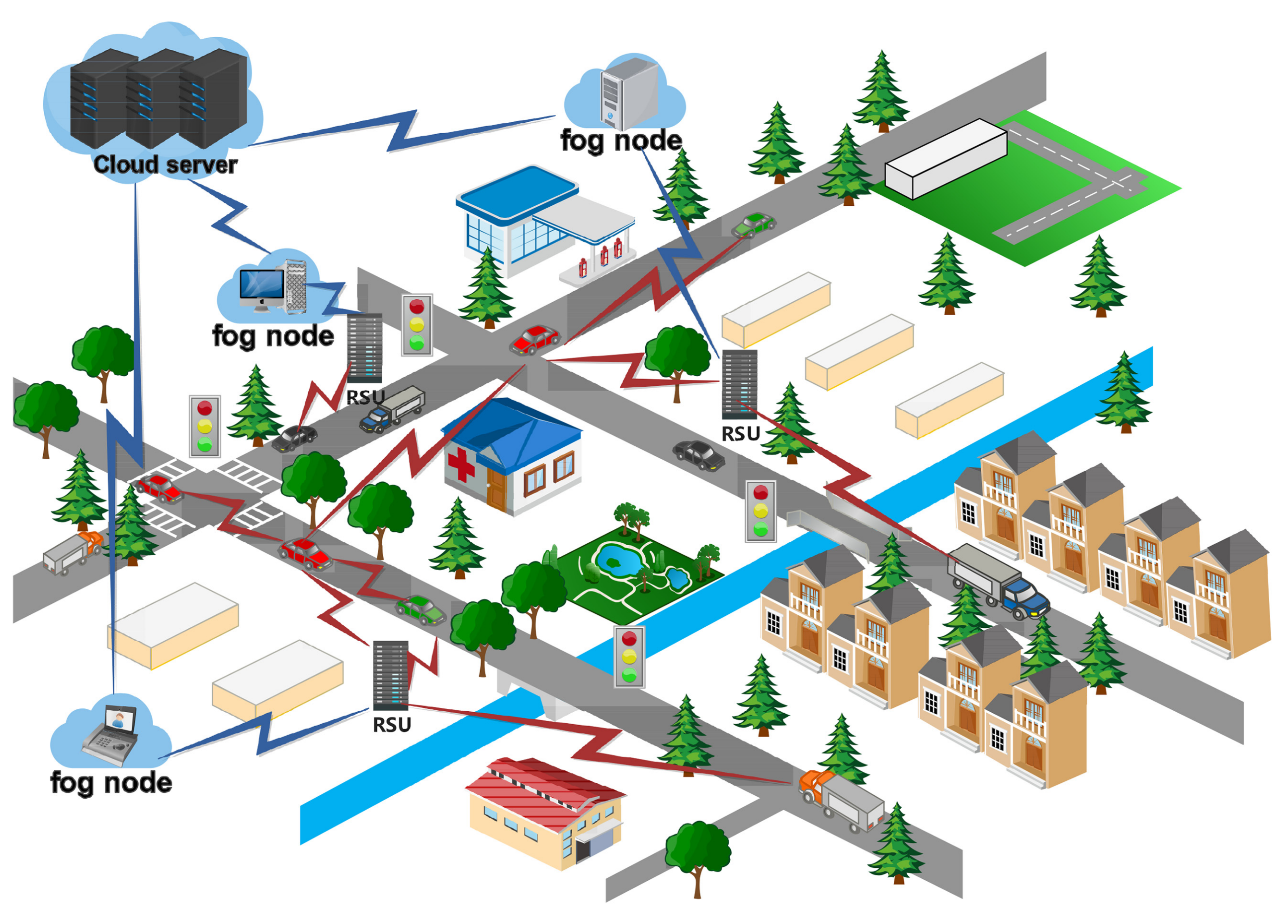

Figure 1. Typical architecture of SIoV.

2. We use elliptic curve algorithms to encrypt the transmission of information, which can provide a higher level of security. We use the Real-or-Random (ROR) model, a formal proof tool, to verify the validity, correctness, and security of the protocol. In addition, a detailed informal analysis shows that our protocol is resistant to known attacks and break-ins.
3. We also systematically evaluate the protocol's computational performance and communication costs in addition to other factors, and show that it performs well.

The rest of this article is organized as follows. Section "Related work" presents the related work of this article. Section "Review of Wu et al.'s protocol" briefly describes Wu et al.'s authentication scheme, followed by a thorough cryptanalysis of Wu et al.'s scheme in section "Cryptanalysis of Wu et al.'s protocol." In section "Cryptanalysis of Wu et al.'s protocol," we propose a new scheme to improve the shortcomings of the old scheme. In section "Security analysis," we perform a security analysis, which includes a formal analysis, security requirements analysis, and a security comparison, to demonstrate the security and stability of the new protocol in terms of these three aspects. In section "Performance evaluation," we analyze the security and performance of the new protocol in terms of both performance evaluation and communication cost evaluation. Finally, we summarize the work in section "Conclusion."

Related work

With the advancement of the application of IoT, IoT technology is combined with social networks to form a new network, that is, SIoT. With the combination of the IoV and SIoT, the social IoV has gradually emerged. Because the IoT environment, IoV environment, and SIoT environment have been proposed, many researchers have attempted studying how to realize data transmission safely and efficiently. Therefore, various authentication protocols have been proposed to protect the security and privacy of data transmissions.

In 2014, Yang et al.[1] presented an abstract network model for IoV, described the technologies needed to create IoV, and different applications based on existing technologies, presented several open research challenges, and considered the development of IoV in future domains. In 2015, Sun et al.[2] reviewed IoV-related security and privacy developments, including security and privacy requirements, types of attacks and solutions, and described the future IoV-related security and privacy developments and challenges. In 2017, Contreras-Castillo et al.[3] presented IoV-related architectures, protocols, and security and introduced communication protocols that enable seamless integration and operation of IoV. Dandala et al.[4] described the relation between the IoV environment and traffic management, and provided an IoV-based traffic management solution to overcome serious traffic management problems in real-life. Ferrag et al.[5] provided an overview of the previously proposed IoV-related protocols and classified these protocols according to the target environment, identified remaining issues, and proposed future directions for the research. In 2019, Chandrakar et al.[6] proposed a secure authentication protocol for vehicle ad hoc networks and claimed that the protocol is secure and efficient. In 2020, Xu et al.[7] proposed a blockchain-based protocol for RSU-assisted authentication and key management in vehicle networks, which they claimed to have low computational overhead, high efficiency, high authentication efficiency, and resistance to various common attacks.

Some of the research on SIoT is presented below. In 2011, Atzori et al.[8] presented the research concept of SIoT and a preliminary architecture for achieving SIoT in an object structure that follows the definition of potential social responsibility. In 2012, Atzori et al.[9] again presented the concept, architecture, and network of SIoT and analyzed the characteristics of the SIoT network structure through simulations. In 2015, Nitti et al.[10] discussed the link selection problem in SIoT, proposed heuristic algorithms for local link selection, and proposed a method to dynamically adjust the threshold of the number of connections according to the number of hubs in the network. In 2017, Shen et al.[11] proposed a privacy-preserving and lightweight key negotiation protocol based on V2G in social IoT and claimed that the protocol can withstand different types of attacks. In 2019, Park et al.[12] proposed a V2G dynamic privacy-preserving key management protocol for the SIoT and claimed that the protocol is resistant to a variety of attacks, such as simulations and offline passwords.

This final section presents the research work related to SIoV. In 2015, Alam et al.[13] presented concepts, structures, and applications of the architecture of SIoV environments and provided implementation details and experimental analysis to demonstrate the effectiveness of the proposed system. In 2016, Maglaras et al.[14] combined SIoV with smart cities, reviewing SIoV enabling technologies and key components, and presenting SIoV applications that can be deployed in smart cities. In 2018, Butt et al.[15] presented a scalable SIoV architecture based on the *Restful web* technology and highlighted the importance of *web* technology. In 2020, Ahmed et al.[16] proposed an anonymous key negotiation protocol for the V2G environment in SIoV and claimed that the protocol is not only lightweight, but also efficient in terms of communication and storage costs of other protocols. In 2021, Wu et al.[17] proposed a lightweight authentication key negotiation protocol for vehicular social networks based on fog nodes and claimed the protocol to be lightweight, secure, and efficient.

Review of Wu et al.'s protocol

The main entities included in the protocol are the vehicle, fog node, and cloud server. A fog node can detect unsafe driving behavior in real-time, provide early warning for the behavior, impose appropriate penalty when necessary, and share the pressure of the cloud server. Table 1 lists the symbols used in the protocol. The protocol has three phases as follows: vehicle registration, fog node registration, and login authentication.

Vehicle registration phase

The registration process of the vehicle V_i is described as follows:

1. First, vehicle V_i inputs its identity ID_i, password PSW_i, and a random number r_i, calculates its pseudo-identity $PID_i = h(ID_i \parallel r_i)$, and then transmits the PID_i to CS through the secure channel.
2. CS receives $\{PID_i\}$, calculates the value of $HID_i = h(PID_i \parallel K_{CS})$, initializes the value of K_V

Table 1. Notations used in Wu et al.'s 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

to 0, and stores $\{PID_i, K_V\}$ in its database. Finally, CS sends $\{HID_i, K_V\}$ to V_i.

3. V_i receives $\{HID_i, K_V\}$. Using HID_i, PSW_i, r_i, and ID_i, it calculates the value $\alpha_i = HID_i \oplus h(PSW_i \parallel r_i), P_i = h(ID_i \parallel PSW_i \parallel r_i)$, replaces HID_i with the value of α_i, and stores the $\{\alpha_i, P_i, r_i, K_V\}$ in its smart card.

Fog node registration phase

The registration process of the FN_j is described as follows:

1. First, fog node FN_j inputs its identity FID_j and a random number r_j, by FID_j and r_j, calculates its pseudo-identity $PFID_j = h(FID_j \parallel r_j)$, and sends $\{PFID_j, FID_j\}$ to CS.
2. CS receives $\{PFID_j, FID_j\}$, selects a random number R_j, calculates the value of $N_j = h(FID_j \parallel ID_{CS}) \oplus R_j$, $K_{FN} = h(PFID_j \parallel K_{CS})$, and $HID_j = h(FID_j \parallel K_{CS})$, and stores $\{PFID_j, K_{FN}, FID_j\}$ in its database. Finally, CS sends $\{K_{FN}, HID_j, N_j, ID_{CS}\}$ to FN_j.
3. FN_j receives $\{K_{FN}$, HID_j, N_j, $ID_{CS}\}$, calculates the value $R_j = h(FID_j \parallel ID_{CS}) \oplus N_j$ and $\beta_j = HID_j \oplus h(R_j \parallel r_j)$, and stores the $\{K_{FN}, \beta_j, r_j, N_j\}$ in its database.

Login and authentication phase

In the login and authentication phase, V_i, FN_j, and CS complete authentication and establish session key SK, which is described as shown in Figure 2.

1. First, V_i inputs its identity ID_i, password PSW_i, according to ID_i, PSW_i, and r_i, calculates $P_i^* = h(ID_i \parallel PSW_i \parallel r_i)$, and then compares $P_i^* \overset{?}{=} P_i$. If equal, then V_i logs successfully. After successful login, V_i selects a random number N_1 and calculates $A_1 = h(ID_i \parallel r_i) \oplus N_1$, $HID_i = \alpha_i \oplus h(PSW_i \parallel r_i)$, and $V_1 = h(HID_i \parallel K_V) \oplus N_1$. Finally, V_i sends the login request $M_1 = \{A_1, V_1, ID_{CS}, PID_i\}$ to FN_j through a common channel.
2. FN_j receives $\{A_1, V_1, ID_{CS}, PID_i\}$, selects a random number N_2, according to A_1, K_{FN}, HID_j, and N_2, calculates $A_2 = h(A_1 \parallel K_{FN} \parallel HID_j) \oplus N_2$, $V_2 = h(A_2 \parallel K_{FN} \parallel V_1)$, and finally FN_j sends $M_2 = \{PID_i, PFID_j, A_2, V_1, V_2\}$ to CS.
3. After CS receives $\{PID_i, PFID_j, A_2, V_1, V_2\}$, indexes K_{FN} according to $FPID_j$, then calculates $HID_i = h(PID_i \parallel K_{CS})$, $N_1 = h(HID_i \parallel K_V) \oplus V_1$, $V_1^* = h(HID_i \parallel K_V) \oplus N_1$, checks $V_1^* \overset{?}{=} V_1$. If it is equal, then V_i is legal. Otherwise, the authentication process is terminated. CS calculates $V_2^* = h(A_2 \parallel K_{FN} \parallel V_1)$ and compares $V_2^* \overset{?}{=} V_2$. 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 $A_1 = N_1 \oplus PID_i$, $HID_j = h(FID_j \parallel K_{CS})$, $N_2 = h(A_1 \parallel K_{FN} \parallel HID_j) \oplus A_2$, selects a random number N_3, and calculates $N_X' = h(HID_i \parallel N_1) \oplus N_2 \oplus N_3 \oplus HID_j$, $N_Y' = h(HID_j \parallel N_2) \oplus N_1 \oplus N_3 \oplus HID_i$, $SK = h(N_1 \oplus N_2 \oplus N_3 \oplus HID_j \oplus HID_i)$, $V_3 = h(HID_j \parallel K_{FN} \parallel SK)$, $V_4 = h(HID_i \parallel K_V \parallel SK)$, then updates $K_V = K_V + 1$, and sends message $M_3 = \{N_X', N_Y', V_3, V_4\}$ to FN_j.
4. FN_j receives $\{N_X', N_Y', V_3, V_4\}$, calculates $N_1 \oplus N_3 \oplus HID_i = h(HID_j \parallel N_2) \oplus N_Y'$, $SK = h(N_1 \oplus N_2 \oplus N_3 \oplus HID_j \oplus HID_i)$, and $V_3^* = h(HID_j \parallel K_{FN} \parallel SK)$, and checks $V_3^* \overset{?}{=} V_3$. 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. V_i receives $\{N_X', V_4\}$, then calculates $N_2 \oplus N_3 \oplus HID_j = = h(HID_i \parallel N_1) \oplus N_X'$, $SK = h(N_1 \oplus N_2 \oplus N_3 \oplus HID_j \oplus HID_i)$, $V_4^* = h(HID_i \parallel K_V \parallel SK)$, and checks $V_4^* \overset{?}{=} V_4$. 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 $K_V = K_V + 1$.

Cryptanalysis of Wu et al.'s protocol

This section focuses on various security flaws in the attacker model, Wu et al.'s protocol. Wu et al. claimed that it is secure against common attacks and is safe and efficient. However, we show that Wu et al.'s protocol does not resist insider attacks and smart card theft attacks and does not ensure perfect forward security.

Threat model

In this study, we define a potential attacker as $\mathcal{A}$. He may be an external attacker who listens to or intercepts data, or a staff member or privileged user inside the server or fog node. When $\mathcal{A}$ acts as an external attacker, he can eavesdrop and intercept messages in the public channel without being detected by the subject protocol, can send or forge messages, and can participate in the operation of the protocol as a legitimate protocol participant. This is partially similar to the capabilities of the attacker assumed by the $D-Y$ model. When $\mathcal{A}$ acts as an insider attacker, he may have some privilege to access parts of the server or fog node as part of the system participants. Based on existing research, we assume that $\mathcal{A}$ has the following capabilities:

1. $\mathcal{A}$ can eavesdrop and intercept information transmitted through the public channel, and can forge, modify, delete, redirect, or replay messages transmitted through the public channel.[18]

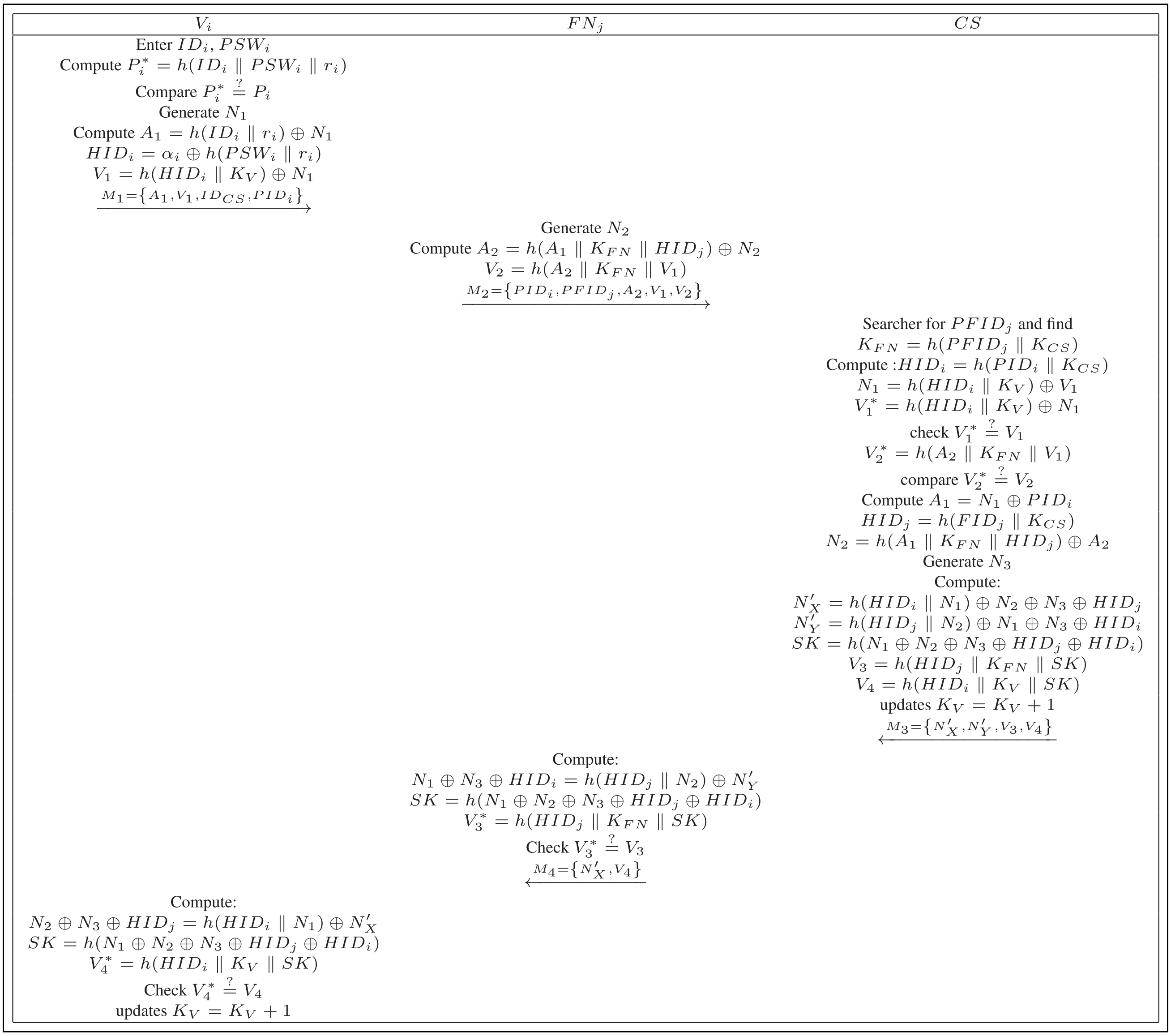

Figure 2. Login and authentication phase.

2. When a smart card or vehicle is lost or stolen, $\mathcal{A}$ can obtain the parameters and useful information that is stored in a smart card or vehicle.[19]
3. $\mathcal{A}$ may be a legitimate but malicious administrator or privileged user.[20]

Insider attack

In an insider attack, the server can also be used as an attacker to steal user information, for example, by collecting the identifier and password submitted by the user during the registration phase and by collecting information from the user's smart card.[21]

Assuming that $\mathcal{A}$ is an internal person, he can obtain the information stored in the smart card $\{\alpha_i, P_i, r_i, K_V\}$. The attacker can guess the password repeatedly, calculate the authentication value, and complete the password guessing through the following steps:

Step 1: $\mathcal{A}$ intercepts the information A_1 and PID_i transmitted to the common channel, and then calculates that N_1 passes $A_1 = N_1 \oplus PID_i$.
Step 2: because $\mathcal{A}$ obtains A_1, r_i, and N_1, $\mathcal{A}$ can try to enter the value of ID_i to calculate $A_1^* = h(ID_i \parallel r_i) \oplus N_1$.
Step 3: $\mathcal{A}$ compares and verifies the calculated A_1^* with A_1 to obtain ID_i.
Step 4: after $\mathcal{A}$ obtains ID_i, $\mathcal{A}$ also knows P_i, thus, he calculates $P_i^* = h(ID_i \parallel PSW_i \parallel r_i)$ and verifies $P_i^* = P_i$. If the verification is successful, $\mathcal{A}$ obtains ID_i and PSW_i.

Therefore, attacker can complete the password guessing.

Lack of perfect forward security

Forward security means that the leakage of a long-used master key does not lead to the leakage of a past session key SK. Forward security protects communications performed in the past from the threat of future exposure of passwords or keys.[22,23]

We assume that the attacker can steal $M_1 = \{A_1, V_1, ID_{CS}, PID_i\}$, $M_2 = \{PID_i, PFID_j, A_2, V_1, V_2\}$, and $M_3 = \{N'_X, N'_Y, V_3, V_4\}$ in the login and mutual authentication phases because they are transmitted over a common channel. The attacker can calculate session key SK using the following steps:

Step 1: the attacker can obtain K_{CS} by the first attack, and then obtain HID_i by calculating $h(PID_i \parallel K_{CS})$.
Step 2: obtain N_1 by calculating $A_1 \oplus PID_i$.
Step 3: obtain $(N_2 \oplus N_3 \oplus HID_j)$ by calculating $h(HID_i \parallel N_1) \oplus N'_X$.

Therefore, the attacker can calculate the correct session key $SK = h(N_1 \oplus N_2 \oplus N_3 \oplus HID_j \oplus HID_i)$.

Smart card theft attack

A smart card theft attack occurs when secret information stored on a smart card is obtained by some unethical means, and the attacker uses the information obtained to crack the session key or cause damage to the protocol.[23]

We assume that the attacker steals the smart card and obtains $\{\alpha_i, P_i, r_i, K_V\}$. The attacker can steal $M_1 = \{A_1, V_1, ID_{CS}, PID_i\}$, $M_2 = \{PID_i, PFID_j, A_2, V_1, V_2\}$, and $M_3 = \{N'_X, N'_Y, V_3, V_4\}$ in the login and mutual authentication phase because they are transmitted over a common channel. The attacker can calculate session key SK using the following steps:

Step 1: the attacker can obtain PSW_i by the first attack, and then obtain HID_i by calculating $\alpha_i \oplus h(PSW_i \parallel r_i)$.
Step 2: obtain N_1 by calculating $A_1 \oplus PID_i$.
Step 3: obtain $(N_2 \oplus N_3 \oplus HID_j)$ by calculating $h(HID_i \parallel N_1) \oplus N'_X$.

Therefore, the attacker can calculate the correct session key $SK = h(N_1 \oplus N_2 \oplus N_3 \oplus HID_j \oplus HID_i)$.

The proposed protocol

In this section, we elaborate the various components of the protocol. First, the protocol involves three constituent entities as follows: (1) the vehicle V_i, (2) the fog node F_j, and (3) the cloud server CS. V_i can establish a session key SK with the cloud server via the fog node, and then CS can exchange information to obtain useful information, such as real-time road conditions and weather conditions. F_j is the equivalent of a trusted intermediary between V_i and CS, which verifies the legitimacy of CS, accepts authentication and requests from V_i and sends them to CS or receives feedback from CS and sends them to V_i. CS has the function of processing data, saving and transmitting information, and it plays an important role in the protocol. CS registers the legal identity of V_i and F_j in the registration phase and provides legal authentication and key establishment for V_i and F_j in the authentication phase. The protocol consists of the following parts: (1) vehicle registration phase, (2) fog node registration phase, and (3) login and mutual authentication phase. Table 2 lists the symbols used in the protocol.

Table 2. Notations used in the improved protocol.

Symbol	Description
V_i	The *i*th vehicle
F_j	The *j*th fog node
CS	Cloud server
VID_i, VPW_i	Identities of V_i, password of V_i
FID_j	Identities of F_j
K_{fc}	Shared key of F_j and CS
K_C	Secret key of CS
SK	Session key
T_i	The *i*th timestamp
$h(\cdot)$	Hash function
$\oplus$	Bit-wise XOR operation
$\parallel$	Concatenate operation

Vehicle registration phase

In the V_i registration phase, V_i sends the registration request to the CS over a secure channel, and the CS then computes a series of messages and returns them to V_i, allowing V_i to obtain a legitimate identity. The process diagram for this phase is shown in Figure 3, and the steps are detailed as follows:

1. First, V_i selects and enters his identity VID_i and password VPW_i, and then V_i transmits $\{VID_i\}$ to the CS via a secure channel.
2. Following receipt of the information from V_i, CS generates the random number r_2 and computes $HID_i = h(VID_i \parallel r_2)$ and $RID_i = h(K_c \parallel r_2) \oplus HID_i$, and then stores $\{RID_i, r_2\}$ in its own memory and subsequently transmits the random number $\{r_2\}$ to V_i via a secure channel.
3. Following receipt of the information from CS, V_i generates a random number r_1 and then computes $P_i = h(VID_i \oplus VPW_i \parallel r_1)$ and $A_1 = h(VPW_i \parallel r_1) \oplus r_2$ and stores $\{A_1, P_i, r_1\}$ in

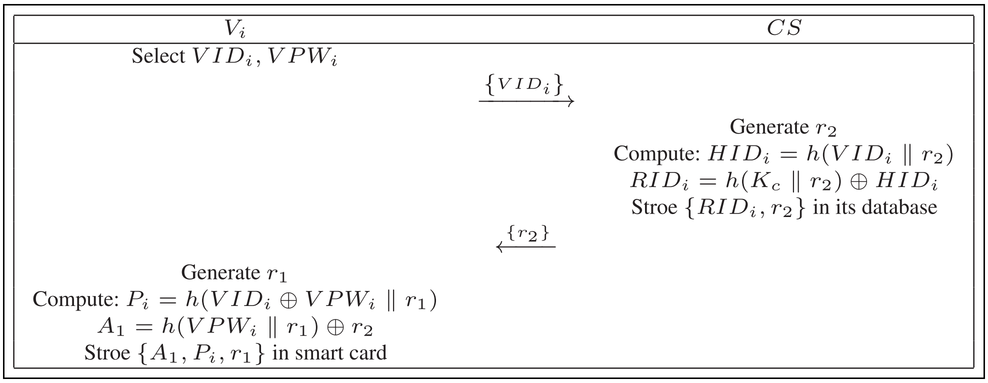

Figure 3. V_i registration phase.

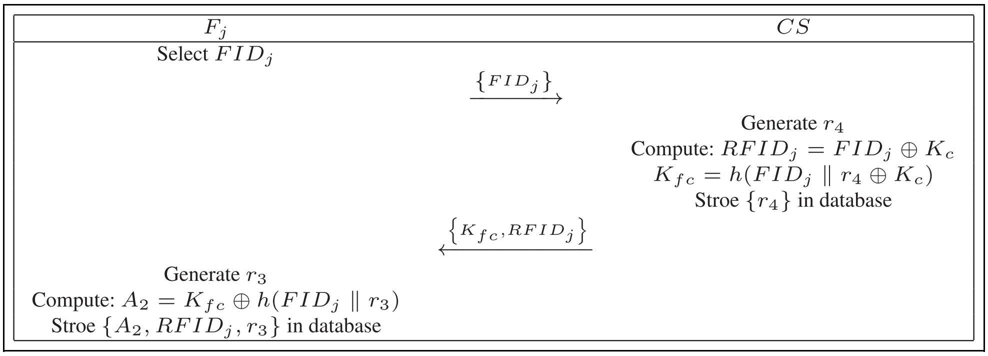

Figure 4. F_j registration phase.

the smart card. The V_i registration phase is complete.

Fog node registration phase

In preparation for the authentication phase, F_j sends a registration request to the CS and registers as a legitimate fog node. The detailed process diagram of this phase is shown in Figure 4, and the detailed steps are as follows:

1. F_j selects a unique identity FID_j and then transmits $\{FID_j\}$ to the CS through a secure channel.
2. After receiving the message from F_j, CS generates a random number r_4 and calculates $RFID_j = FID_j \oplus K_c$ and $K_{fc} = h(FID_j \parallel r_4 \oplus K_c)$. Then, CS stores r_4 into memory according to its $RFID_j$ counterpart and subsequently transmits the message $\{K_{fc}, RFID_j\}$ to F_j through a secure channel.
3. Once F_j receives the information from CS, he generates the random number r_3 and then starts computing $A_2 = K_{fc} \oplus h(FID_j \parallel r_3)$, preferably storing $\{A_2, RFID_j, r_3\}$ in his own memory. This completes the F_j registration phase.

Login and authentication phase

In the login and mutual authentication phase, the on-board login device verifies the correctness of the identifiers VID_i^* and VPW_i^* entered by V_i, and only those V_i that pass the verification will be allowed to use the system. During the mutual authentication phase, V_i, F_j, and CS negotiate a common session key SK to allow for quick information sharing during subsequent use. This phase is the most important stage of the protocol, and the detailed process is described in Figure 5, and the detailed steps are described as follows:

1. First, the on-board device verifies the correctness and legitimacy of the user, V_i inputs VID_i^* and VPW_i^*, calculates $P_i^* = h(VID_i^* \parallel VPW_i^* \parallel r_1)$ and then verifies that $P_i^* \overset{?}{=} P_i$. If they are equal, authentication is successful; otherwise, login is denied.
2. After completing verification, V_i computes $r_2 = A_1 \oplus h(VPW_i \parallel r_1)$ and $HID_i = h(VID_i \parallel r_2)$, and then generates a random number R_1 and timestamp T_1. It encapsulates R_1 into B_1 by computing $B_1 = h(HID_i \parallel r_2) \oplus R_1$ and then computes $V_1 = h(HID_i \parallel r_2)$ and subsequently transmits the

V_i	F_j	CS
Imput VID_i^*, VPW_i^* Compute: $P_i^* = h(VID_i^* \parallel VPW_i^* \parallel r_1)$ Check: $P_i^* \overset{?}{=} P_i$ $r_2 = A_1 \oplus h(VPW_i \parallel r_1)$ $HID_i = h(VID_i \parallel r_2)$ Generate R_1, T_1 Compute: $B_1 = h(HID_i \parallel r_2) \oplus R_1$ $V_1 = h(HID_i \parallel r_2)$ $\xrightarrow{Msg_1=\{B_1,V_1,T_1\}}$		
	Verify $\lvert T_1 - T_c \rvert \overset{?}{\leqq} \Delta T$ Generate R_2, T_2 Compute: $K_{fc} = A_2 \oplus h(FID_j \parallel r_3)$ $B_2 = h(K_{fc} \parallel FID_j) \oplus R_2$ $V_2 = h(FID_j \oplus K_{fc})$ $\xrightarrow{Msg_2=\{RFID_j,B_1,B_2,V_1,V_2,T_2\}}$	
		Verify $\lvert T_2 - T_c \rvert \overset{?}{\leqq} \Delta T$ K_{fc} is calculated from FID_j Compute: $FID_j = RFID_j \oplus K_c$ $K_{fc} = h(FID_j \parallel r_4 \oplus K_c)$ $HID_i = RID_i \oplus h(K_c \parallel r_2)$ $V_1^* = h(HID_i \parallel r_2)$ $V_2^* = h(FID_j \oplus K_{fc})$ Check: $V_1^* \overset{?}{=} V_1, V_2^* \overset{?}{=} V_2$ $R_1 = B_1 \oplus h(HID_i \parallel r_2)$ $R_2 = B_2 \oplus H(K_{fc} \parallel FID_j)$ Generate R_3, T_3 $SK = h(HID_i \oplus FID_j \oplus R_2 \oplus R_3 \parallel R_1)$ $B_3 = h(FID_j \oplus K_{fc}) \oplus (HID_i \oplus R_3 \parallel R_1)$ $B_4 = h(HID_i \oplus r_2) \oplus R_2 \oplus R_3 \oplus FID_j$ $B_5 = h(FID_j \oplus K_{fc}) \oplus h(HID_i \parallel r_2 \oplus R_1)$ $V_3 = h(K_{fc} \parallel FID_j \oplus R_2)$ $\xleftarrow{Msg_3=\{B_3,B_4,B_5,V_3,T_3\}}$
	Verify $\lvert T_3 - T_c \rvert \overset{?}{\leqq} \Delta T$ Compute: $V_3^* = h(K_{fc} \parallel FID_j \oplus R_2)$ Check: $V_3^* \overset{?}{=} V_3$ $HID_i \oplus R_3 \parallel R_1 = B_3 \oplus h(FID_j \oplus K_{fc})$ $SK = h(FID_j \oplus R_2 \oplus HID_i \oplus R_3 \parallel R_1)$ Generate T_4 $V_4 = B_5 \oplus h(FID_j \oplus K_{fc})$ $\xleftarrow{Msg_4=\{B_4,V_4,T_4\}}$	
Verify $\lvert T_4 - T_c \rvert \overset{?}{\leqq} \Delta T$ Compute: $V_4^* = h(HID_i \parallel r_2 \oplus R_1)$ Check: $V_4^* \overset{?}{=} V_4$ $SK = h(B_4 \oplus h(HID_i \oplus r_2) \oplus HID_i \parallel R_1)$		

Figure 5. Login and authentication phase.

message $Msg_1 = \{B_1, V_1, T_1\}$ to F_j through the common channel.

3. Immediately after receiving the message Msg_1 from V_i, F_j verifies the timestamp parameter T_1 by computing $\lvert T_1 - T_c \rvert \overset{?}{\leqq} \Delta T$, then generates the random number R_2 and timestamp T_2, computes $K_{fc} = A_2 \oplus h(FID_j \parallel r_3)$, $B_2 = h(K_{fc} \parallel FID_j) \oplus R_2$, and $V_2 = h(FID_j \oplus K_{fc})$, and finally the message $Msg_1 = \{RFID_j, B_1, B_2, V_1, V_2, T_2\}$ is transmitted to the CS via the common channel.
4. After receiving the message Msg_2 from F_j, CS verifies the timestamp T_2 and calculates FID_j, $FID_j = RFID_j \oplus K_c$, $K_{fc} = h(FID_j \parallel r_4 \oplus K_c)$, $HID_i = RID_i \oplus h(K_c \parallel r_2)$, $V_1^* = h(HID_i \parallel r_2)$, and $V_2^* = h(FID_j \oplus K_{fc})$, and then verifies that $V_1^* \overset{?}{=} V_1$ to authenticate the legitimacy and validity of V_i's identity by verifying that $V_2^* \overset{?}{=} V_2$ to determine the legitimacy of the identity of F_j. Then, CS computes $R_1 = B_1 \oplus h(HID_i \parallel r_2)$ and $R_2 = B_2 \oplus H(K_{fc} \parallel FID_j)$ to generate a random number R_3 and timestamp T_3. Once this is complete, CS generates the session key SK, $SK = h(HID_i \oplus FID_j \oplus R_2 \oplus R_3 \parallel R_1)$. Then calculate $B_3 = h(FID_j \oplus K_{fc}) \oplus (HID_i \oplus R_3 \parallel R_1)$, $B_4 = h(HID_i \oplus r_2) \oplus R_2 \oplus R_3 \oplus FID_j$, $B_5 = h(FID_j \oplus K_{fc}) \oplus h(HID_i \parallel r_2 \oplus R_1)$, and $V_3 = h(K_{fc} \parallel FID_j \oplus R_2)$, and then the message $Msg_3 = \{B_3, B_4, B_5, V_3, T_3\}$ is sent to F_j through the common channel.
5. After F_j receives the message Msg_3, it starts verifying timestamp T_3, and if T_3 passes the verification,

the nominal message Msg_3 is considered to be a new and valid message. F_j then computes $V_3^* = h(K_{fc} \| FID_j \oplus R_2)$ and verifies that $V_3^* \overset{?}{=} V_3$. If the verification passes, CS is a trusted server; otherwise, F_j rejects the CS request and aborts the protocol process. If it passes, F_j computes $HID_i \oplus R_3 \| R_1 = B_3 \oplus h(FID_j \oplus K_{fc})$ and $SK = h(FID_j \oplus R_2 \oplus HID_i \oplus R_3 \| R_1)$, and subsequently generates timestamp T_4, computes $V_4 = B_5 \oplus h(FID_j \oplus K_{fc})$, and transmits the message $Msg_4 = \{B_4, V_4, T_4\}$ to V_i through the common channel.

6. V_i receives the message Msg_4 back from F_j and verifies the freshness and legitimacy of this message by $|T_4 - T_c| \overset{?}{\leqq} \Delta T$, and then verifies the legitimacy identity of F_j by computing $V_4^* = h(HID_i \| r_2 \oplus R_1)$. If the authentication passes, V_i computes the session key $SK = h(B_4 \oplus h(HID_i \oplus r_2) \oplus HID_i \| R_1)$. By completing the aforementioned steps, the login and authentication phase of the protocol is complete, and a common session key SK is established between the three parties V_i, F_j, and CS.

Security analysis

In this section, a formal security analysis, an analysis of security requirements, and a security comparison are performed to demonstrate the security of our proposed scheme. First, the formal analysis uses the real-or-random (ROR) model, and then the analysis of security requirements demonstrates that our proposed protocol is resistant to insider attacks, smart card theft attacks, and ensures perfect forward security. Finally, by comparing the security of our protocol with that of Ma et al.,[24] Jia et al.,[25] Eftekhari et al.,[26] and Wu et al.,[17] we can observe that our protocol is secure and reliable.

Formal security analysis

In this section, the ROR model[27] is used to perform a formal security analysis. The ROR model is used to prove the semantic security of the proposed protocol. Using the ROR model, we successfully proved that the session key of the protocol is secure and reliable. Before proving the session key security of the proposed protocol in Theorem 1, we briefly discuss the ROR model.

ROR model. In our ROR model, the attacker is represented by $\mathcal{A}$, and the protocol has three participants: the vehicle, fog node, and cloud server and are represented by V, F, and CS, respectively. Assuming that Γ_{all} denotes the communication between $\mathcal{A}$ and the protocol entity, then Γ_V^i denotes that $\mathcal{A}$ communicates with the ith instance of the vehicle, Γ_F^j denotes that $\mathcal{A}$ communicates with the jth instance of the fog node, and Γ_{CS} denotes that $\mathcal{A}$ communicates with the cloud server. The attacker $\mathcal{A}$ can also obtain relevant information through the following queries:

Execute$(\Gamma_V^i, \Gamma_F^j, \Gamma_{CS}^k)$, where $\mathcal{A}$ can intercept and obtain information exchanged or transmitted between communicating entities V, F, and CS through the open channel. This query is often used to perform eavesdropping attacks.

Send(Γ_{all}, Msg): using this query, $\mathcal{A}$ can send a message Msg to any entity in Γ_{all} and obtain the corresponding feedback. $\mathcal{A}$ can perform man-in-the-middle and simulated attacks.

Hash(*String*): in this query, $\mathcal{A}$ can obtain the corresponding fixed value after executing the query by entering a fixed-length string.

Corrupt(Γ_{all}): $\mathcal{A}$ can send this query to Γ_V^i and fetch the private value stored in the smart card of V_i. Furthermore, $\mathcal{A}$ can send this query to Γ_F^j or Γ_{CS}, which then obtains the long-term private key stored in the cloud server and the temporary information generated by the participant. $\mathcal{A}$ can perform forward secrecy attacks, privileged insider attacks, stolen smart card attacks, and vehicle simulation attacks with this query.

Reveal(Γ_{all}): using this query, $\mathcal{A}$ can disclose the session key SK generated between Γ_{all} entities to $\mathcal{A}$. $\mathcal{A}$ can then simulate the known session key to perform the attack.

Test(Γ_{all}): $\mathcal{A}$ can perform this query by flipping a uniformly textured coin $\bigcirc$. If $\bigcirc$ is 1, the attacker will obtain the correct session key. Otherwise, the attacker will receive a null value.

Theorem 1: if $Adv_{mathcalA}^{AKE}(xi)$ is a function of the dominance of adversary *mathcalA* in breaching the *SK* security of the proposed authenticated key exchange (AKE) protocol, then q_{hans} and q_{send} denote the number of *hash* queries performed and the *send* queries performed, respectively. f denotes the length of a user's identity as well as the password, C' and b' denote the parameters of Zipf,[28] and then

$$Adv_{\mathcal{A}}^{AKE}(\xi) \leqslant 2max\left\{\frac{C' \cdot q_{send}^{b'}, q_{send}}{2^f}\right\} + \frac{q_{send}}{2^{f-2}} + \frac{3q_{hash}^2}{2^{f-1}} \tag{1}$$

Security proof

Proof. In the following proof, we define six games named GM(i), $i \in [0, 6]$, and each game has its own rule. We define $Succ_{\mathcal{A}}^{GM_i}(\xi)$ ($i = 0, 1, 2, 3, 4, 5, 6$) to represent the probability of success of the game under each rule. In addition, "$\mathcal{A}$'s advantage in winning a match GM_i” is expressed and defined by $Adv_{A,GM_i}^{AKE}(\xi)$. The specific proof procedure is as follows:

GM_0: in GM_0, this round simulates $\mathcal{A}$ for the actual attack, and because the bit $\bigcirc$ is selected randomly at the start of GM_0, we obtain

$$Adv_{\mathcal{A}}^{AK\mathcal{E}}(\xi) = \left|2Adv_{A,GM_0}^{AKE}(\xi) - 1\right| \tag{2}$$

GM_1: GM_1 adds the *Execute* operation to GM_0, which is equivalent to $\mathcal{A}$ intercepting and obtaining information on the public channel $\{Msg_1, Msg_2, Msg_3, Msg_4\}$, and $\mathcal{A}$ executes the *Test* operation, thus, we obtain

$$Adv_{A,GM_1}^{AKE}(\xi) = Adv_{A,GM_0}^{AKE}(\xi) \tag{3}$$

GM_2: GM_2 adds the *Send* operation to GM_1, and $\mathcal{A}$ can send messages to the entity through the common channel, thus, we can obtain

$$\left|Adv_{A,GM_2}^{AKE}(\xi) - Adv_{A,GM_1}^{AKE}(\xi)\right| \leqslant \frac{q_{send}}{2^f} \tag{4}$$

GM_3: GM_3 adds another *Hash* operation to GM_2, and $\mathcal{A}$ can use *hash* queries to obtain specific values and strings. Using the theory of the birthday paradox, we obtain

$$\left|Adv_{A,GM_3}^{AKE}(\xi) - Adv_{A,GM_2}^{AKE}(\xi)\right| \leqslant \frac{q_{hash}^2}{2^{f+1}} \tag{5}$$

GM_4: in GM_4, we have added the partial functionality of the *Corrupt* operation to GM_3 that allows $\mathcal{A}$ to obtain the long-term key K_{fc} between CS and F_j or to crack any random number in the protocol authentication process. Under these conditions, we consider the $\mathcal{A}$ threats to the session key SK, verifying that the protocol has perfect forward security and is resistant to known session-specific temporary information attacks.

1. Perfect forward secrecy: we assume $\mathcal{A}$ uses *Corrupt* queries to obtain the long-term key K_c, and then $\mathcal{A}$ uses *Execute*, *Send*, *Hash*, and *Corrupt* operations to attempt to obtain the protocol's session key SK. After $\mathcal{A}$ obtains K_c, $\mathcal{A}$ can obtain $RFID_j$ in the message Msg_2 on the public channel using the *Execute* operation, and then $FID_j = RFID_j \oplus K_c$ to compute FID_j. If $\mathcal{A}$ computes K_{fc}, $\mathcal{A}$ can compute R_2 by $R_2 = B_2 \oplus H(K_{fc} \parallel FID_j)$. Then, $HID_i \oplus R_3 \parallel R_1 = B_3 \oplus h(FID_j \oplus K_{fc})$ computes $HID_i \oplus R_3 \parallel R_1$ to compute SK. Thus, everything points to K_{fc}, however, as $K_{fc} = h(FID_j \parallel r_4 \oplus K_c)$, $\mathcal{A}$ cannot obtain r_4; therefore, he cannot compute K_{fc}, and cannot threaten the protocol SK.
2. Known session-specific temporary information attacks: we assume $\mathcal{A}$ uses the *Corrupt* query to obtain a random number R_2 that is most likely to crack SK, and then $\mathcal{A}$ uses the *Execute* operation to obtain the information B_2 and B_3 on the common channel. Subsequently, $\mathcal{A}$ can calculate $h(K_{fc} \parallel FID_j) = B_2 \oplus R_2$ to obtain $h(K_{fc} \parallel FID_j)$, and then calculate $B_3 = h(FID_j \oplus K_{fc}) \oplus (HID_i \oplus R_3 \parallel R_1)$ to obtain $(HID_i \oplus R_3 \parallel R_1)$. However, $\mathcal{A}$ cannot compute or intercept the acquisition of FID_j; thus, $\mathcal{A}$ cannot threaten the protocol SK. As a result, the probability of this round is

$$\begin{aligned}&\left|Adv_{A,GM_4}^{AKE}(\xi) - Adv_{A,GM_3}^{AKE}(\xi)\right|\\ &\leqslant \frac{q_{hash}^2}{2^{f+1}} + \frac{q_{send}}{2^f}\end{aligned} \tag{6}$$

GM_5: in GM_5, we have added additional parts of the *Corrupt* operation to GM_4 to allow $\mathcal{A}$ to access the information stored in the smart card via V_i to verify that the protocol is resistant to offline password guessing attacks. We assume that $\mathcal{A}$ has access to the information stored on the smart card $\{A_2, RFID_j, r_3\}$, because $\mathcal{A}$ has no other useful information about V_i, $\mathcal{A}$ cannot decrypt the information about V_i, thus, cannot compute the session key SK. Using Zipf's law,[28] the probability that $\mathcal{A}$ succeeds in guessing the user's password is 1/2, and the probability that $\mathcal{A}$ can successfully guess the user's password is greater than 1/2 when the number of bits transmitted ends $\leqslant 106$. Thus, we obtain

$$\begin{aligned}&\left|Adv_{A,GM_5}^{AKE}(\xi) - Adv_{A,GM_4}^{AKE}(\xi)\right|\\ &\leqslant max\left\{C' \cdot q_{send}^{b'}, \tfrac{q_{send}}{2^f}\right\}\end{aligned} \tag{7}$$

GM_6: GM_6 is used to verify that the proposed protocol is resistant to simulation attacks. In GM_6, $\mathcal{A}$ issues a $h(FID_j \oplus R_2 \oplus HID_i \oplus R_3 \parallel R_1)$ query to determine whether it is possible to obtain SK. Here, the game was aborted. Thus, we can obtain the possibility of GM_6 as

$$\left|Adv_{A,GM_6}^{AKE}(\xi) - Adv_{A,GM_5}^{AKE}(\xi)\right| \leqslant \frac{q_{hash}^2}{2^{f+1}} \tag{8}$$

Because GM_6 has an equal probability of success and failure, the

$$Adv_{A,GM_6}^{AKE}(\xi) = \frac{1}{2} \tag{9}$$

From the aforementioned formula above, we can obtain

$$\begin{aligned}\frac{1}{2}Adv_{\mathcal{A}}^{AKE}(\xi) &= \left|Adv_{\mathcal{A},\,GM_0}^{AKE}(\xi) - \frac{1}{2}\right| \\ &= \left|Adv_{\mathcal{A},\,GM_0}^{AKE}(\xi) - Adv_{\mathcal{A},\,GM_6}^{AKE}(\xi)\right| \\ &= \left|Adv_{\mathcal{A},\,GM_1}^{AKE}(\xi) - Adv_{\mathcal{A},\,GM_6}^{AKE}(\xi)\right| \\ &\leqslant \sum_{i=1}^{6}\left|Adv_{\mathcal{A},\,GM_i}^{AKE}(\xi) - Adv_{\mathcal{A},\,GM_{i-1}}^{AKE}(\xi)\right| \\ &= max\left\{C' \cdot q_{send}^{b'}, \frac{q_{send}}{2^f}\right\} \\ &+ \frac{q_{send}}{2^{f-1}} + \frac{3q_{hash}^2}{2^f}\end{aligned} \tag{10}$$

Then, we obtain

$$\begin{aligned}Adv_{\mathcal{A}}^{AKE}(\xi) \leqslant 2max\left\{C' \cdot q_{send}^{b'}, \frac{q_{send}}{2^f}\right\} \\ + \frac{q_{send}}{2^{f-2}} + \frac{3q_{hash}^2}{2^{f-1}}\end{aligned} \tag{11}$$

Thus, we can use the ROR model to demonstrate that our proposed new protocol is resistant to common attacks (such as smart card theft attacks, offline password guessing attacks, man-in-middle attacks, and known session-specific temporary information attacks) and provides perfect forward security.

Analysis of security requirements

This section presents an analysis of our security requirements for the proposed protocol, which shows that our protocol can withstand attacks that the protocol proposed by Wu et al.[17] cannot, as well as other common attacks. In the following, we use $\mathcal{A}$ to represent the attacker, as demonstrated by the following:

Resist insider attacks.

Assuming $\mathcal{A}$ obtains V_i in the smart card $\{A_1, P_i, r_1\}$, he can attempt to compute $P_i^* = h(VID_i^* \parallel VPW_i^* \parallel r_1)$. However, guessing both VID_i and VPW_i is nearly impossible, and $\mathcal{A}$ would be unable to obtain the user's identifier and password by collecting information from the user. Thus, our protocol is resistant to internal attacks.

Ensure perfect forward secrecy.

In the protocol, assuming that the long-term key K_c of CS is compromised, $\mathcal{A}$ can obtain FID_j by calculating $FID_j = RFID_j \oplus K_c$ because $RFID_j$ is a public channel transmission. However, $\mathcal{A}$ cannot calculate K_{fc}. This is because in $K_{fc} = h(FID_j \parallel r_4 \oplus K_c)$, $\mathcal{A}$ cannot obtain the value of r_4 and cannot compute useful concrete information. Thus, our protocol has a perfect forward security.

Resist stolen smart card attacks.

Assuming that $\mathcal{A}$ obtains the information in V_i's smart card $\{A_1, P_i, r_1\}$. Because $\mathcal{A}$ cannot obtain V_i's identifier VID_i and password VPW_i, $\mathcal{A}$ cannot decrypt the relevant information about V_i, and thus, cannot compute the session key SK. Therefore, our protocol is resistant to stolen smart card attacks.

Ensure mutual authentication.

During the login authentication phase, V_i, F_j, and CS can authenticate each other and establish the same session key in a secure manner. The V_1 in the Msg_1 message contains information about V_i. F_j receives Msg_1 and encapsulates V_1 and its own information V_2 in Msg_2 and transmits it to CS, which authenticates V_i and F_j by verifying V_1 and V_2. F_j can achieve authentication of CS by verifying V_3 in the message Msg_3, and V_i achieves authentication of F_j by verifying V_4 in message Msg_4. Thus, mutual authentication is ensured among the three participants in our protocol.

Ensure user anonymity.

In the protocol, we do not use V_i's real identity VID_i but a pseudo-identity HID_i, and no information related to V_i's identity is transmitted on the public channel which effectively protects user privacy. If $\mathcal{A}$ wants to trace V_i, the timestamped validation also prevents $\mathcal{A}$ from using expired feedback to obtain useful information about the user. Thus, our protocol ensures user anonymity.

Resist replay attacks.

Replay attacks can occur during any network communication and are one of the common attacks used by hackers in the computer world. It refers to the attacker sending a packet that has already been received by the destination host for the purpose of spoofing the system, and is mainly used in the authentication process to undermine the accuracy of the authentication. In our protocol, we add timestamps T to all messages $\{Msg_1, Msg_2, Msg_3, Msg_4\}$, to ensure the timeliness and freshness of the transmitted information, to ensure that the transmission of the message is completed within a valid time, and to prevent the attacker from replaying the message to obtain valid feedback. Thus, our protocol can resist replay attacks.

Resist offline password guessing attacks.

In the login and authentication phase, V_i must enter both VID_i^*, and VPW_i^*, and then compute $P_i^* = h(VID_i^* \parallel VPW_i^* \parallel r_1)$ when logging in. Even if $\mathcal{A}$ obtains the information r_1 in the smart card, it cannot guess both VID_i and VPW_i; thus, $\mathcal{A}$ cannot obtain V_i's identifier and password through the guessing attack.

Resist known session-specific temporary information attacks.

During the login authentication phase, three random numbers are generated: R_1, R_2, and R_3. These three random numbers are also part of the session key. Assuming that $\mathcal{A}$ learns the random number R_1, he can only obtain $h(HID_i \parallel r_2)$ by computing

$B_1 = h(HID_i \parallel r_2) \oplus R_1$ and nothing else. Assuming that $\mathcal{A}$ learns the random number R_2, because B_2 and B_3 are transmitted on a common channel, $\mathcal{A}$ can obtain $h(K_{fc} \parallel FID_j)$ by computing $R_2 = B_2 \oplus h(K_{fc} \parallel FID_j)$, and then can obtain $HID_i \oplus R_3 \parallel R_1$ by computing $HID_i \oplus R_3 \parallel R_1 = B_3 \oplus h(FID_j \oplus K_{fc})$. However, $\mathcal{A}$ cannot obtain FID_j, and therefore cannot compute SK. We assume that $\mathcal{A}$ learns the random number R_3; however, he cannot compute useful information. Therefore, our protocol is resistant to known session-speculative temporary information attacks.

Resist man-in-the-middle attacks. A man-in-the-middle attack is performed by intercepting normal network communication data and performing data tampering and sniffing without the knowledge of the two parties communicating. In the framework environment, F_j does not authenticate V_i but sends its own authentication information along with that of V_i to CS, which promptly authenticates V_i and F_j. If $\mathcal{A}$ tampers with the data during the process, it will be subjected to a double test of the timestamp and CS authentication. Clearly, $\mathcal{A}$ will not be able to pass authentication safely and will be denied access. Therefore, our protocol is resistant to man-in-the-middle attacks.

Security comparisons

As shown in Table 3, we compare the security analysis of the protocol and use ✓ and × to indicate whether the protocol meets the relevant security requirements.

As shown in the table, the protocol of Ma et al.[24] is considered by Eftekhari et al.[26] to be unable to resist insider attacks, provide anonymity and untraceability, and resist known session-specific temporary information attacks and stolen smart card/vehicle attacks. Furthermore, the protocol of Jia et al.[25] cannot provide mutual authentication and cannot resist known session-specific temporary information attacks. Therefore, in 2021, Eftekhari et al.[26] proposed a security-enhanced three-party pairwise shared key agreement protocol for fog-based vehicle communication. They claimed that they can save approximately 23.65% of the computing costs. However, the protocol of Eftekhari et al.[26] cannot guarantee perfect forward secrecy. In addition, Wu et al.[17] proposed a lightweight authentication key protocol based on a fog node in SIoV. In this study, we demonstrated that it cannot guarantee perfect forward security and cannot resist insider attacks and stolen smart card attacks.

Performance evaluation

In this section, we compare the performance of the proposed protocol with the protocol in Table 3, which includes calculation evaluation and communication evaluation. In terms of computing evaluation, we used more real simulation experiments. The use of mobile phones and computers to simulate an environment can more accurately reflect the computing performance of the protocol.

Hardware environment

We used the mobile phone MEIZU – MX5 to simulate the on-board equipment, the computer model Lenovo – M715E to simulate the fog node, and the computer model MSI – GP63 to simulate the cloud server. Table 4 shows the platform used for the equipment.

Computation evaluation

Based on the aforementioned platform, we also calculated the following cryptographic operations according to the time consumption: hash function, point encryption, symmetric key encryption/decryption, scalar multiplication, and binary pairing. Here, the time consumption of the XOR operation and connection operation is very small to be ignored, and the abbreviations and consumption times corresponding to various operations are shown in Table 5.

Table 3. Comparisons of security.

Security properties	Ma et al.[24]	Jia et al.[25]	Eftekhari et al.[26]	Wu et al.[17]	Ours
Resist insider attacks	×	✓	✓	×	✓
Ensure perfect forward secrecy	✓	✓	×	×	✓
Resist stolen smart card attacks	×	✓	✓	×	✓
Ensure mutual authentication	✓	×	✓	✓	✓
Ensure user anonymity	×	✓	✓	✓	✓
Resist replay attacks	✓	✓	✓	✓	✓
Resist offline password guessing attacks	✓	✓	✓	✓	✓
Known session-specific temporary information attacks	×	×	✓	✓	✓
Resist man-in-the-middle attacks	✓	✓	✓	✓	✓

Table 4. Simulation platform.

Device	MEIZU – MX5	Lenovo – M715E	MSI – GP63
Operating system	Flyme 6.3.5.0A	Windows 10	Windows 10
CPU	Helio X10 Turbo	Pentium(R)CPU E5500@2.80 GHz	Intel(R) i7-8750HCPU@2.20 GHz
Memory	3 GB RAM	2 GB RAM	24 GB RAM

CPU: central processing unit; RAM: random access memory.

Table 5. Execution time of basic operation.

Operations	Abbreviation	MEIZU – MX5 (ms)	Lenovo – M715E (ms)	MSI – GP63 (ms)
Hash function	T_h	0.0049	0.0044	0.0025
Point addition	T_{ad}	0.4894	0.1723	0.0527
Encryption/decryption	T_{ed}	17.213	11.477	8.094
Scala multiplication	T_{sm}	7.983	5.889	3.221
Bilinear pairing	T_{bp}	21.607	15.532	8.607

Table 6. Computation cost comparison.

Protocol	V_i	F_j	CS	Total (ms)
Ma et al.[24]	$4T_h + 3T_{sm}$	$4T_h + 4T_{sm}$	$11T_h + 10T_{sm}$	$19T_h + 17T_{sm} \approx 79.7797$
Jia et al.[25]	$6T_h + 2T_{sm} + 1T_{bp}$	$4T_h + 2T_{sm} + 1T_{bp}$	$11T_h + 3T_{sm} + 1T_{bp}$	$21T_h + 7T_{sm} + 3T_{bp} \approx 83.2275$
Eftekhari et al.[26]	$11T_h + 3T_{sm} + 1T_{ad}$	$12T_h + 3T_{sm} + 1T_{ad}$	$15T_h + 3T_{sm} + 2T_{ad}$	$38T_h + 9T_{sm} + 4T_{ad} \approx 52.1903$
Wu et al.[17]	$7T_h$	$5T_h$	$11T_h$	$23T_h \approx 0.0838$
Ours	$8T_h$	$7T_h$	$11T_h$	$26T_h \approx 0.0975$

To evaluate the calculation cost of the protocol, we divide the time cost of each protocol into four parts: V_i, F_j, CS, and the total calculation cost, and calculate the time spent in each part to more accurately reflect the performance of the protocol. The specific calculation costs are shown in Table 6. After a detailed comparison, we can observe that the time cost of our protocol is similar to that of Wu et al.;[17] however, our protocol provides higher security and reliability. Compared with Ma et al.,[24] Jia et al.,[25] and Eftekhari et al.,[26] the proposed protocol is much faster and saves considerable computing costs. In addition to saving costs, our protocol can ensure high security, while requiring less time.

Communication evaluation

In terms of computation cost evaluation, we define the output of the hash function to account for 160 bits, the random/non-random number as 160 bits, the elliptic curve points as 320 bits, the identifier as 64 bits, and the timestamp as 32 bits. The message sent by V_i in our protocol is $Msg_1 = \{B_1, V_1, T_1\}$ and the communication cost is [160 + 160 + 32], the message sent by F_j is $Msg_2 = \{RFID_j, B_1, B_2, V_1, V_2, T_2\}$ and $Msg_4 = \{B_4, V_4, T_4\}$ and the communication cost is [160 + 160 + 160 + 160 + 160 + 160 + 160 + 32 + 160 + 160 + 32], and the CS sends a message with $Msg_3 = \{B_3, B_4, B_5, V_3, T_3\}$ with a communication cost of [160 + 160 + 160 + 160 + 160 + 32], adding up to a total cost of 2208 bits. After our calculation of the message data size transmitted by the protocol in Figure 6 at each stage, the total calculation is shown in Figure 6. At stage V_i, our protocol spends the least amount of communication, imposing the least amount of computational stress on the vehicle user. In stage F_j, our fog node computational pressure is not significantly different from Wu et al.'s[17] protocol; however, it is much better than other protocols and can reduce communication costs. For the cloud server, the communication cost of our protocol is the same as that of Jia et al.[25] and is not much different from that of Wu et al.[17] in terms of overall communication cost, and our protocol is the least expensive in terms of communication cost, which is less than half of that of Ma et al.[24] In short, although our protocol has a negligible difference in computational cost compared to Wu et al.'s protocol, we are better than Wu et al.'s protocol in terms of communication cost, not to mention that our protocol has better security than Wu et al.'s protocol and can withstand attacks that Wu et al. cannot. All things considered, our protocol is very efficient and secure.

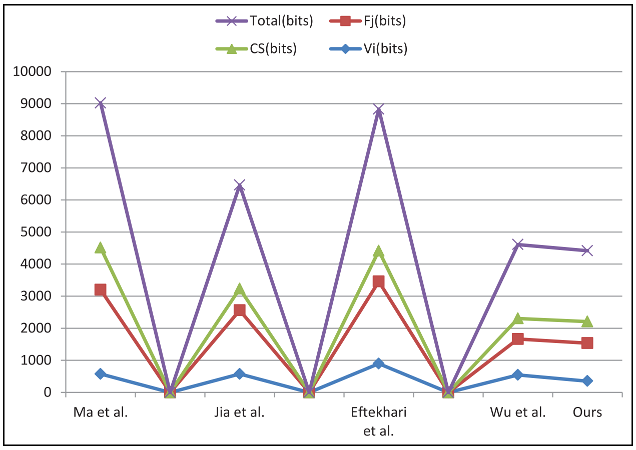

Figure 6. Communication cost evaluation.

Conclusion

In this study, we improved the protocol proposed by Wu et al. in social telematics. The improved protocol is a fast and secure authentication protocol based on the fog node that operates in the SIoV, which does not ensure perfect forward security and is not resistant to insider and smart card theft attacks. The improved protocol not only compensates for the vulnerabilities and flaws of the existing protocol and can successfully resist attacks that the original protocol cannot, but can also resist replay attacks, insider attacks, simulated attacks, and more aggressive known session-specific temporary information attacks. It also exhibits excellent performance and efficiency in terms of security and computational cost. Therefore, it can be considered more suitable for use in fog-based SIoV. Contemporary research needs to address not only connected vehicle problems, but also some ancillary classes of problems, such as high precision maps. Currently, there are technical challenges for high precision maps, as well as policy and regulatory challenges, and this aspect is beyond the scope of this article.

In the future, SIoV will become a new starting point and a new pursuit for IoV development. SIoV will help vehicles become fully intelligent and greatly improve the user's travel experience. We should be thankful that we live in an era of rapid social change, and I hope this article will provide a reference to address the security of SIoV data.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Chien-Ming Chen https://orcid.org/0000-0002-6502-472X

References

1. Yang F, Wang S, Li J, et al. An overview of Internet of vehicles. *China Commun* 2014; 11(10): 1–15.
2. Sun Y, Wu L, Wu S, et al. Security and privacy in the Internet of vehicles. In: *2015 international conference on identification, information, and knowledge in the Internet of Things (IIKI)*, Beijing, China, 22–23 October 2015, pp.116–121. New York: IEEE.
3. Contreras-Castillo J, Zeadally S and Guerrero-Ibañez JA. Internet of vehicles: architecture, protocols, and security. *IEEE Internet Things* 2017; 5(5): 3701–3709.
4. Dandala TT, Krishnamurthy V and Alwan R. Internet of vehicles (IoV) for traffic management. In: *2017 international conference on computer, communication and signal processing (ICCCSP)*, Chennai, India, 10–11 January 2017, pp.1–4. New York: IEEE.
5. Ferrag MA, Maglaras LA, Janicke H, et al. Authentication protocols for internet of things: a comprehensive survey. *Secur Commun Netw* 2017; 2017: 6562953.
6. Chandrakar P, Jain A, Balivada S, et al. A secure authentication protocol for vehicular ad-hoc networks. In: *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*, Coimbatore, India, 20–22 February 2019, pp.1–7. New York: IEEE.
7. Xu Z, Liang W, Li KC, et al. A blockchain-based roadside unit-assisted authentication and key agreement protocol for Internet of vehicles. *J Parallel Distr Com* 2021; 149: 29–39.
8. Atzori L, Iera A and Morabito G. SIoT: giving a social structure to the internet of things. *IEEE Commun Lett* 2011; 15(11): 1193–1195.
9. Atzori L, Iera A, Morabito G, et al. The social Internet of things (SIoT)–when social networks meet the internet of things: concept, architecture and network characterization. *Comput Netw* 2012; 56(16): 3594–3608.
10. Nitti M, Atzori L and Cvijikj IP. Friendship selection in the social Internet of things: challenges and possible strategies. *IEEE Internet Things* 2014; 2(3): 240–247.
11. Shen J, Zhou T, Wei F, et al. Privacy-preserving and lightweight key agreement protocol for V2G in the social Internet of things. *IEEE Internet Things* 2017; 5(4): 2526–2536.
12. Park K, Park Y, Das AK, et al. A dynamic privacy-preserving key management protocol for V2G in social Internet of things. *IEEE Access* 2019; 7: 76812–76832.
13. Alam KM, Saini M and El Saddik A. Toward social Internet of vehicles: concept, architecture, and applications. *IEEE Access* 2015; 3: 343–357.
14. Maglaras LA, Al-Bayatti AH, He Y, et al. Social Internet of vehicles for smart cities. *J Sens Actuator Netw* 2016; 5(1): 3.
15. Butt TA, Iqbal R, Shah SC, et al. Social internet of vehicles: architecture and enabling technologies. *Comput Electr Eng* 2018; 69: 68–84.
16. Ahmed S, Kumari S, Saleem MA, et al. Anonymous key-agreement protocol for V2G environment within social internet of vehicles. *IEEE Access* 2020; 8: 119829–119839.

17. Wu TY, Guo X, Yang L, et al. A lightweight authenticated key agreement protocol using fog nodes in social Internet of vehicles. *Mob Inf Syst* 2021; 2021: 3277113.
18. Dolev D and Yao A. On the security of public key protocols. *IEEE T Inform Theory* 1983; 29(2): 198–208.
19. Messerges TS, Dabbish EA and Sloan RH. Examining smart-card security under the threat of power analysis attacks. *IEEE T Comput* 2002; 51(5): 541–552.
20. Azam F, Yadav SK, Priyadarshi N, et al. A comprehensive review of authentication schemes in vehicular ad-hoc network. *IEEE Access* 2021; 9: 31309–31321.
21. Chen CM, Li Z, Chaudhry SA, et al. Attacks and solutions for a two-factor authentication protocol for wireless body area networks. *Secur Commun Netw* 2021; 2021: 3116593.
22. Wu F, Li X, Xu L, et al. A novel three-factor authentication protocol for wireless sensor networks with IoT notion. *IEEE Syst J* 2020; 15(1): 1120–1129.
23. Masud M, Gaba GS, Choudhary K, et al. Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare. *IEEE Internet Things* 2021; 9: 2649–2656.
24. Ma M, He D, Wang H, et al. An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks. *IEEE Internet Things* 2019; 6(5): 8065–8075.
25. Jia X, He D, Kumar N, et al. Authenticated key agreement scheme for fog-driven IoT healthcare system. *Wirel Netw* 2019; 25(8): 4737–4750.
26. Eftekhari SA, Nikooghadam M and Rafighi M. Security-enhanced three-party pairwise secret key agreement protocol for fog-based vehicular ad-hoc communications. *Veh Commun* 2021; 28: 100306.
27. Abdalla M, Fouque PA and Pointcheval D. Password-based authenticated key exchange in the three-party setting. In: *International workshop on public key cryptography*, Les Diablerets, 23–26 January 2005, pp.65–84. Berlin: Springer.
28. Wang D, Cheng H, Wang P, et al. Zipf's law in passwords. *IEEE T Inf Foren Sec* 2017; 12(11): 2776–2791.