


Article

# Secure Authentication Protocol for Wireless Sensor Networks in Vehicular Communications

SungJin Yu <sup>1</sup> , JoonYoung Lee <sup>1</sup>, KyungKeun Lee <sup>2</sup>, KiSung Park <sup>1,\*</sup> and YoungHo Park <sup>1,\*</sup>

<sup>1</sup> School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea; darkskiln@naver.com (S.Y.); harry250@naver.com (J.L.)

<sup>2</sup> Samsung Electronics, Suwon 16677, Korea; crypto.knu@gmail.com

\* Correspondence: kisung2@ee.knu.ac.kr (K.P.); parkyh@knu.ac.kr (Y.P.); Tel.: +82-53-950-7842 (Y.P.)

Received: 27 July 2018; Accepted: 18 September 2018; Published: 21 September 2018



**Abstract:** With wireless sensor networks (WSNs), a driver can access various useful information for convenient driving, such as traffic congestion, emergence, vehicle accidents, and speed. However, a driver and traffic manager can be vulnerable to various attacks because such information is transmitted through a public channel. Therefore, secure mutual authentication has become an important security issue, and many authentication schemes have been proposed. In 2017, Mohit et al. proposed an authentication protocol for WSNs in vehicular communications to ensure secure mutual authentication. However, their scheme cannot resist various attacks such as impersonation and trace attacks, and their scheme cannot provide secure mutual authentication, session key security, and anonymity. In this paper, we propose a secure authentication protocol for WSNs in vehicular communications to resolve the security weaknesses of Mohit et al.'s scheme. Our authentication protocol prevents various attacks and achieves secure mutual authentication and anonymity by using dynamic parameters that are changed every session. We prove that our protocol provides secure mutual authentication by using the Burrows–Abadi–Needham logic, which is a widely accepted formal security analysis. We perform a formal security verification by using the well-known Automated Validation of Internet Security Protocols and Applications tool, which shows that the proposed protocol is safe against replay and man-in-the-middle attacks. We compare the performance and security properties of our protocol with other related schemes. Overall, the proposed protocol provides better security features and a comparable computation cost. Therefore, the proposed protocol can be applied to practical WSNs-based vehicular communications.

**Keywords:** authentication; wireless sensor network; vehicular communications; formal security analysis; BAN logic; AVISPA

## 1. Introduction

Wireless sensor networks (WSNs), in conjunction with intelligent transport systems (ITS) and embedded technology, have advanced to such an extent that drivers can make full use of various information such as traffic congestion, vehicle accidents, and speed. To provide these useful services, a sensor in the vehicle collects data on the vehicle and surrounding area and sends it to the traffic manager through a sink node. The traffic manager in the traffic management office receives data from vehicle sensors and can monitor a vehicle and the surrounding area to provide useful data to the driver in real time. However, a malicious adversary can easily obtain and modify the data because it is transmitted via a public network. Therefore, the authentication protocol between the vehicle and user in vehicular communications has become a very important security issue. In the last few decades, numerous authentication schemes for WSNs have been proposed to ensure secure communications and user privacy [1–8]. In 2006, Wong et al. [9] proposed a dynamic ID-based user

authentication scheme for WSNs. However, Das et al. [10] showed that Wong et al.'s [9] scheme is vulnerable to the stolen verifier attack and proposed an improved two-factor authentication scheme to overcome these security problems. In 2010, Chen et al. [11] demonstrated that Das et al.'s scheme [10] cannot provide secure mutual authentication and cannot resist parallel session attacks. To resolve this problem, they proposed a robust mutual authentication scheme for WSNs. Khan et al. [12] also showed that Das et al.'s scheme [10] cannot prevent the privileged insider and bypassing attacks, nor can it provide mutual authentication and the password changing phase. To overcome these security weaknesses, they proposed a two-factor user authentication protocol that uses secret parameters. In 2011, Yeh et al. [13] found that Das et al.'s scheme cannot resist the insider attack and provide mutual authentication, which are essential security requirements for the WSNs. They proposed a secured authentication protocol for WSNs that uses elliptic curve cryptography (ECC). Unfortunately, Han [14] pointed out that Yeh et al.'s scheme cannot provide mutual authentication, perfect forward secrecy, and key agreement. To resolve the security weaknesses of Yeh et al.'s scheme, Shi et al. [15] proposed a new user authentication protocol for WSNs using ECC. However, Choi et al. [16] showed that Shi et al.'s [15] scheme is vulnerable to a smartcard being stolen, sensor energy exhaustion, and session key attacks. They proposed a new user authentication protocol based on ECC.

In the last few decades, numerous protocols for secure vehicle communications have been proposed [17–25]. In 2008, Zhang et al. [17] proposed an efficient roadside unit (RSU)-aided message authentication scheme that uses a hash message authentication code (HMAC) for vehicular communications networks. Zhang et al. also proposed [18] an efficient message authentication scheme for vehicular communications. Lu et al. [19] proposed an efficient conditional privacy preservation protocol for secure vehicular communications that uses bilinear pairing. However, their protocol is not efficient in resource-constrained vehicular ad hoc networks (VANETs) because it has used multiple anonymous key and has high latency for generating of pseudo-random keys [20]. In 2014, Chuang and Lee [21] proposed an authentication mechanism for vehicle to vehicle communications in VANETs. However, in 2016, Kumari et al. [22] showed that Chuang and Lee's authentication protocol is vulnerable to insider and impersonation attacks, and they proposed an enhanced authentication protocol for VANETs. In 2017, Mohit et al. [23] also proposed an authentication protocol for WSNs in vehicle communications. Mohit et al. claimed that their proposed scheme can resist various attacks such as smartcard stolen, impersonation, and untraceable attacks. In this paper, however, we demonstrate that their scheme cannot resist impersonation and trace attacks. In addition, we show that Mohit et al.'s scheme cannot provide anonymity, session key security, and mutual authentication. We propose a secure authentication protocol for WSNs in vehicle communications that overcomes these security weaknesses.

### 1.1. Threat Model

To analyze the security of our proposed scheme, we introduce the Dolev–Yao (DY) threat model, which is widely used to evaluate the security of a protocol. The detailed assumptions of the DY threat model are as follows:

- An adversary can modify, eavesdrop, insert or delete the transmitted messages over a public channel.
- An adversary can obtain a lost or smartcard stolen, and he/she can also extract the information stored in the smartcard [26,27].
- An adversary can perform various attacks such as impersonation, trace, smartcard stolen, and replay attacks.

### 1.2. Our Contributions

The main contributions of this paper are as follows:

- We demonstrate that Mohit et al.'s scheme is vulnerable to various attacks such as impersonation and trace attacks. In addition, we point out that their scheme cannot provide mutual authentication, session key security and anonymity.
- We propose a secure authentication protocol for WSNs in vehicular communications to resolve these security weaknesses. Our proposed protocol prevents impersonation and trace attacks, and also achieves anonymity, session key security and secure mutual authentication. In addition, the proposed scheme is efficient because it utilizes only hash function and XOR operation in authentication phase.
- We prove that our protocol provides secure mutual authentication by using the broadly accepted Burrows–Abadi–Needham (BAN) logic [28]. We also perform an informal analysis to demonstrate the security of the proposed protocol against various attacks such as impersonation and trace attacks.
- We compare the performance of our scheme against those of related existing schemes and perform a formal security verification by using the widespread Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation software tool.

### 1.3. Paper Outline

The remainder of this paper is organized as follows. In Section 2, we introduce the vehicular communications system model. In Sections 3 and 4, we review Mohit et al.'s authentication scheme and analyze its security weaknesses. In Section 5, we propose a secure authentication protocol for WSNs in vehicular communications to resolve the security problems of their scheme. In Section 6, we present an informal analysis on the security of our protocol and prove that it achieves secure mutual authentication by using BAN logic. In Sections 7 and 8, we present the formal security verification with the AVISPA simulation tool and compare the performance of our protocol with that of related protocols. Finally, we present our conclusions in Section 9.

## 2. System Model

In this section, we introduce a vehicular communication system using WSNs and essential security requirements. There are three entities involved in the vehicular communications system: the vehicle sensor, sink node, and user. The vehicular communications system model is shown in Figure 1.

The vehicular communications system consists of two parts: the WSNs and vehicle and the user and sink node. The vehicle sensor is deployed in the vehicle and collects data on the traffic and surrounding area in real time, which it then sends to the sink node. After receiving the data from the vehicle sensor, the sink node stores it for the user. The user can control the response to traffic jams, speed, and emergency situations based on the data collected by the sink node.

The numerous authentication protocols [29–31] have defined security requirements in order to explain their security goals. Therefore, we also define the essential security requirements to explain and ensure our security goals.

- **Untraceability and anonymity.** In a modern vehicular communication system, user's real identity and location data are very sensitive information. For these reason, an adversary cannot trace a user's location and know the user's real identity to guarantee a privacy of user.
- **Secure mutual authentication.** A secure mutual authentication is known for a essential security requirement in VANETs in order to guarantee that only the legitimate users should access the services and communicate securely with each other [32].
- **Confidentiality.** In our system, the user, sink node, and vehicle sense can freely communicate among themselves through a internet. However, an adversary can try to obtain various pieces of information from users such as traffic congestion, speed, and vehicle accident because it is transmitted in a public channel. Therefore, a confidentiality must be guaranteed and the transmitted data is only known to legitimate user in order to ensure a security.

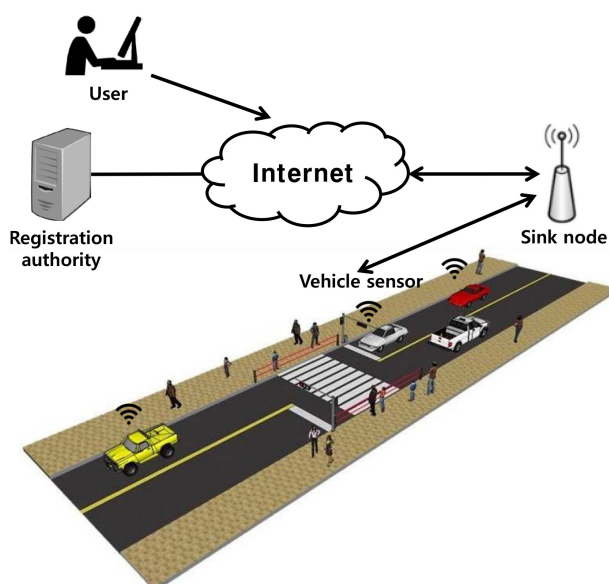


Figure 1. Vehicular communications system model.

### 3. Review of Mohit et al.'s Scheme

In this section, we review Mohit et al.'s authentication protocol for WSNs, which consists of three phases: system setup, user registration, and user login and authentication. Table 1 presents the notations used in this paper.

Table 1. Notations.

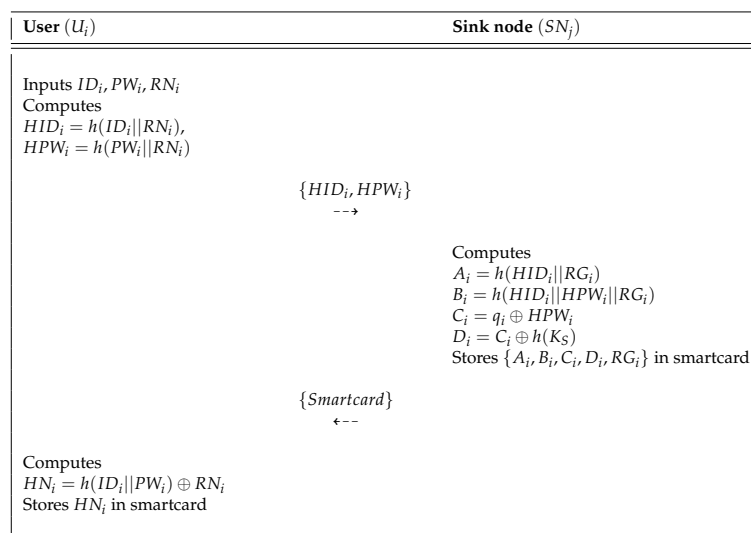
Notation	Description
$ID_i$	Identity of user
$ID_j$	Identity of sink node
$ID_k$	Identity of vehicle sensor
$PW_i$	Password of user
$RA$	Registration authority
$a_i$	Random number by user
$RU_i$	Random nonce by user
$RS_j$	Random nonce by sink node
$RV_k$	Random nonce by vehicle sensor
$K_S$	Master key of sink node
$TID_i$	Unique temporary identity of user
$h(\cdot)$	One-way hash function
$\oplus$	Bitwise XOR operation
$\parallel$	Concatenation operation

#### 3.1. System Setup Phase

When a driver wants to deploy a sensor in a vehicle, the registration authority (RA) registers the vehicle sensor in the network. In addition, RA stores various data on the vehicle such as the vehicle number, engine, battery, and insurance in a database.

### 3.2. User Registration Phase

If a new traffic manager  $U_i$  wants to register him or herself,  $U_i$  must send the registration request message to the sink node  $SN_j$  first. The user registration phase of Mohit et al.'s scheme is shown in Figure 2, and the detailed steps are described as follows.



**Figure 2.** User registration phase of the Mohit et al.'s scheme.

- Step 1:**  $U_i$  chooses an identity  $ID_i$ , password  $PW_i$ , and random nonce  $RN_i$ .  $U_i$  then computes  $HID_i = h(ID_i || RN_i)$ ,  $HPW_i = h(PW_i || RN_i)$  and sends them to the sink node via a secure channel.
- Step 2:**  $SN_j$  selects a random nonce  $RG_i$  and random number  $q_i$ , and then  $SN_j$  computes  $A_i = h(HID_i || RG_i)$ ,  $B_i = h(HID_i || HPW_i || RG_i)$ ,  $C_i = q_i \oplus HPW_i$ , and  $D_i = C_i \oplus h(K_s)$ . After that,  $SN_j$  stores  $\{A_i, B_i, C_i, D_i, RG_i\}$  in the smartcard and issues the smartcard to  $U_i$  through a secure channel.
- Step 3:** Upon receiving the smartcard,  $U_i$  computes  $HN_i = h(ID_i || PW_i) \oplus RN_i$  and stores it in the smartcard. Ultimately, the smartcard contains  $\{A_i, B_i, C_i, D_i, RG_i, HN_i\}$ .

### 3.3. User Login and Authentication Phase

If a user  $U_i$  wants to access the system,  $U_i$  must send the login request message to the sink node  $SN_j$ . After receiving the login request message from  $U_i$ ,  $SN_j$  checks whether it is legitimate. If it is valid,  $SN_j$  performs the authentication phase. The user login and authentication phase of Mohit et al.'s scheme is shown in Figure 3. The detailed steps of this phase are described as follows.

- Step 1:**  $U_i$  inserts the smartcard into a card reader and inputs  $ID_i$  and  $PW_i$ . The smartcard then computes  $RN_i = h(ID_i || PW_i) \oplus HN_i$ ,  $HID_i = h(ID_i || RN_i)$ ,  $HPW_i = h(PW_i || RN_i)$ , and  $B_i^* = h(HID_i || HPW_i || RG_i)$ . Then, the smartcard checks whether  $B_i^* \stackrel{?}{=} B_i$ . If it is equal, the smartcard computes  $q_i = C_i \oplus HPW_i$  and generates a random nonce  $NU_i$ . The smartcard also computes  $M_{TS} = h(q_i || B_i || NU_i)$ ,  $p_1 = NU_i \oplus q_i$ ,  $p_2 = ID_k \oplus h(p_1 || q_i)$  and  $E_i = D_i \oplus HPW_i$ . Finally, the smartcard sends the login request message  $\{M_{TS}, p_1, p_2, E_i\}$  to  $SN_j$  via a public channel.
- Step 2:** After receiving the login request message from  $U_i$ ,  $SN_j$  retrieves  $q_i = E_i \oplus h(K_s)$ ,  $NU_i = p_1 \oplus q_i$  and  $ID_k = p_2 \oplus h(p_1 || q_i)$ . Then,  $SN_j$  computes  $M_{TS}^* = h(q_i || B_i || NU_i)$  and checks whether  $M_{TS}^*$  is equal to  $M_{TS}$ . Then,  $SN_j$  generates a random nonce  $NS_j$  and computes  $X_k = h(ID_k || K_s)$ ,  $M_{SV} = h(ID_k || NS_j || X_k || ID_j)$ ,  $d_1 = NS_j \oplus h(ID_k)$ ,  $d_2 = ID_j \oplus ID_k$ . Finally,  $SN_j$  sends  $\{M_{SV}, d_1, d_2\}$  to the vehicle sensor.

- Step 3:** Upon receiving the message  $\{M_{SV}, d_1, d_2\}$ , the vehicle sensor  $VS_k$  retrieves  $NS_j = d_1 \oplus h(ID_k)$  and  $ID_j = d_2 \oplus ID_k$ . Then,  $VS_k$  checks the freshness of  $NS_j$ . If it is fresh,  $VS_k$  sends  $ID_k$  and requests the sink node's master key  $X_k$  from RA. After receiving  $X_k$  from RA through a secure channel,  $VS_k$  computes  $M_{SV}^* = h(ID_k || NS_j || X_k || ID_j)$  and checks whether  $M_{SV}^* \stackrel{?}{=} M_{SV}$ . If it is verified,  $VS_k$  chooses a random nonce  $NV_k$  and computes  $v = h(ID_k || NS_j || NV_k)$ ,  $M_{VS} = h(X_k || NS_j || v)$ , and  $t = NS_j \oplus NV_k$ . Finally,  $VS_k$  sends  $\{M_{VS}, t\}$  to  $SN_j$ .
- Step 4:** After receiving the message  $\{M_{VS}, t\}$ ,  $SN_j$  retrieves  $NV_k = t \oplus NS_j$  and computes  $v = h(ID_k || NS_j || NV_k)$ ,  $M_{VS}^* = h(X_k || NS_j || v)$ . Then,  $SN_j$  checks whether  $M_{VS}^* \stackrel{?}{=} M_{VS}$  is correct. If it is correct,  $SN_j$  computes  $w = NS_j \oplus NU_i$ ,  $M_{ST} = h(q_i || NU_i || NS_j || ID_i || ID_k)$  and sends  $\{M_{ST}, w\}$  to  $U_i$ .
- Step 5:** Upon receiving the message  $\{M_{ST}, w\}$  from  $SN_j$ ,  $U_i$  retrieves  $NS_j = w \oplus NU_i$  and computes  $M_{ST}^* = h(q_i || NU_i || NS_j || ID_i || ID_k)$ , and then  $U_i$  checks whether  $M_{ST}^* \stackrel{?}{=} h(q_i || NU_i || NS_j || ID_i || ID_k)$  is correct. If they are equal, mutual authentication has been successfully achieved.

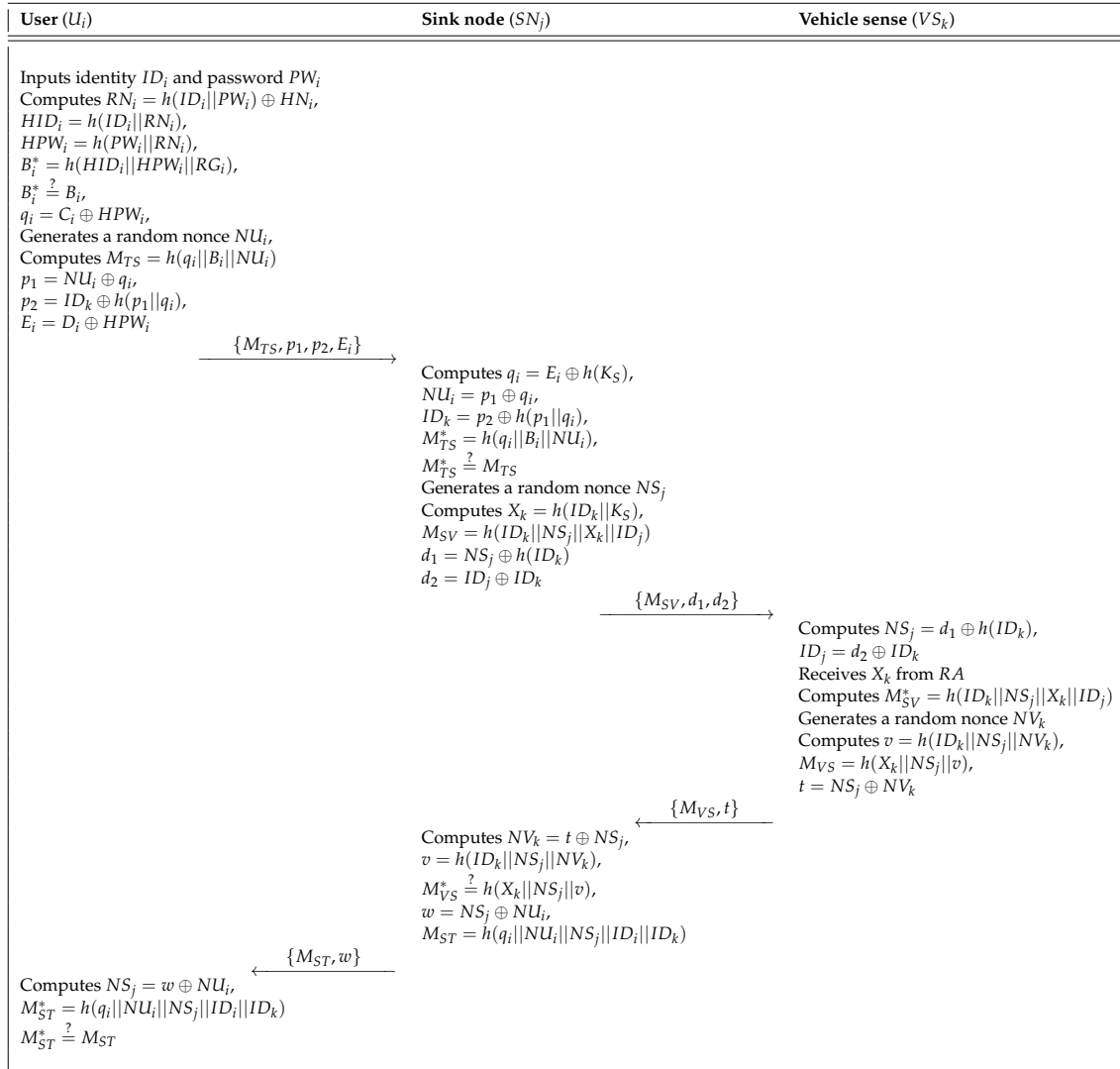
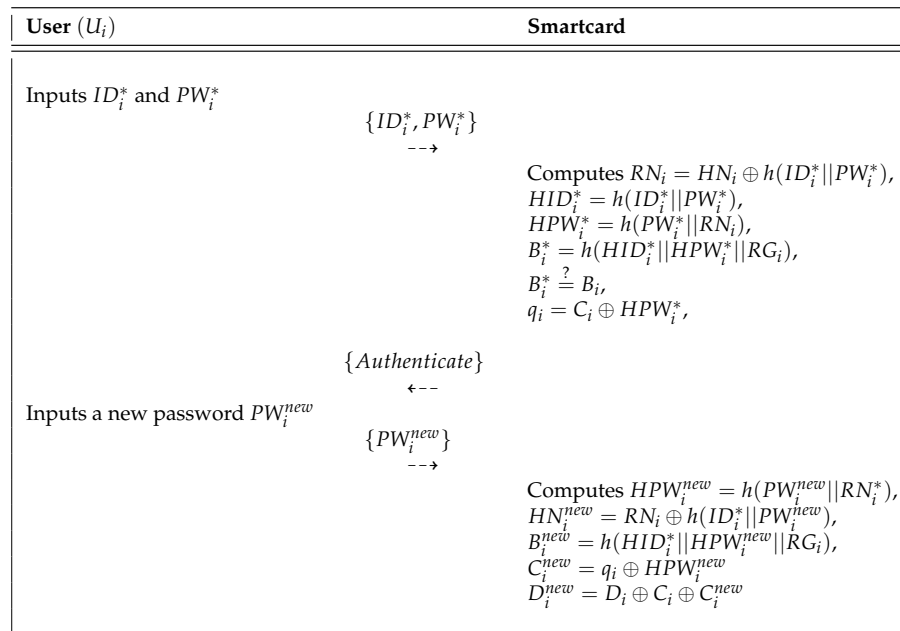


Figure 3. User login and authentication phase of the Mohit et al.'s scheme.

### 3.4. Password Change Phase

$U_i$  can freely update his or her password when desired. The password change phase is described in Figure 4 and the detailed steps of this phase are as follows.



**Figure 4.** Password change phase of the Mohit et al.'s scheme.

- Step 1:**  $U_i$  inserts smartcard in the card reader and inputs the identity  $ID_i^*$  and password  $PW_i^*$ , and then  $U_i$  submits  $\{ID_i^*, PW_i^*\}$  to the card reader via a secure channel.
- Step 2:** After receiving  $\{ID_i^*, PW_i^*\}$ , the smartcard computes  $RN_i = HN_i \oplus h(ID_i^* || PW_i^*)$ ,  $HID_i^* = h(ID_i^* || PW_i^*)$ ,  $HPW_i^* = h(PW_i^* || RN_i)$ , and  $B_i^* = h(HID_i^* || HPW_i^* || RG_i)$ . It checks whether  $B_i^* \stackrel{?}{=} B_i$ . If this is verified, the smartcard sends the authentication message and requests a new password from  $U_i$ . After receiving the authentication message from smartcard,  $U_i$  inputs the new password  $PW_i^{new}$ .
- Step 3:** The smartcard calculates  $HPW_i^{new} = h(PW_i^{new} || RN_i^*)$ ,  $HN_i^{new} = RN_i \oplus h(ID_i^* || PW_i^{new})$ ,  $B_i^{new} = h(HID_i^* || HPW_i^{new} || RG_i)$ ,  $C_i^{new} = q_i \oplus HPW_i^{new}$ , and  $D_i^{new} = D_i \oplus C_i \oplus C_i^{new}$  by using the new password of  $U_i$ . Finally, smartcard replaces  $\{HN_i, B_i, C_i, D_i\}$  with  $\{HN_i^{new}, B_i^{new}, C_i^{new}, D_i^{new}\}$ .

## 4. Cryptanalysis of Mohit et al.'s Scheme

In this section, we discuss the security weaknesses of Mohit et al.'s scheme. They asserted that their scheme is secure against trace and impersonation attack, and they showed that their scheme can provide anonymity, session key security and secure mutual authentication. However, here we demonstrate that Mohit et al.'s scheme does not resist the following attacks.

### 4.1. Impersonation Attack

If an adversary  $U_a$  tries to impersonate a legitimate user,  $U_a$  can successfully generate a login request message of legitimate user  $\{M_{TS}, p_1, p_2, E_i\}$ . According to Section 1.1, we can assume that  $U_a$  obtains the smartcard of the legitimate user  $U_i$  and extracts the values  $\{B_i, C_i, D_i\}$  stored in smartcard and that  $U_a$  has the messages transmitted in the previous session. Here, we show that Mohit et al.'s scheme does not prevent an impersonation attack.



- Step 1:**  $U_a$  computes  $HPW_i = D_i \oplus E_i$ ,  $q_i = C_i \oplus HPW_i$ ,  $NU_i = p_1 \oplus q_i$ ,  $ID_k = p_2 \oplus h(p_1||q_i)$ , and  $M_{TS} = h(q_i||B_i||NU_i)$ , where  $E_i$ ,  $p_1$ , and  $p_2$  are messages of the previous session.
- Step 2:**  $U_a$  can obtain the secret parameters  $q_i$ ,  $B_i$ , and  $HPW_i$  and a random nonce  $NU_i$ .  $U_a$  then chooses a random nonce  $RU_a$  and computes  $M_{TSa} = h(q_i||B_i||NU_a)$ ,  $p_{1a} = NU_a \oplus q_i$ , and  $p_{2a} = ID_k \oplus h(p_{1a}||q_i)$ . Finally,  $U_a$  generates the login request message  $\{M_{TSa}, p_{1a}, p_{2a}, E_i\}$  and sends it to the sink node  $SN_j$ .
- Step 3:** After receiving the login request message from  $U_a$ ,  $SN_j$  retrieves  $q_i = E_i \oplus h(K_s)$ ,  $NU_a = p_{1a} \oplus p_{2a}$ , and  $ID_k = p_{2a} \oplus h(p_{1a}||q_i)$ .  $SN_j$  then computes  $M_{TS}^* = h(q_i||B_i||NU_a)$  and checks whether  $M_{TS}^*$  is equal to  $M_{TSa}$ . Then,  $SN_j$  generates a random nonce  $NS_{j2}$  and computes  $X_k = h(ID_k||K_s)$ ,  $M_{SV2} = h(ID_k||NS_{j2}||X_k||ID_j)$ ,  $d_1 = NS_{j2} \oplus h(ID_k)$ , and  $d_2 = ID_{j2} \oplus ID_k$ . Finally,  $SN_j$  sends  $\{M_{SV2}, d_1, d_2\}$  to the vehicle sensor.
- Step 4:** Upon receiving the message  $\{M_{SV2}, d_1, d_2\}$ , the vehicle sensor  $VS_k$  retrieves  $NS_{j2} = d_1 \oplus h(ID_k)$  and  $ID_j = d_2 \oplus ID_k$ , and then  $VS_k$  checks the freshness of  $NS_{j2}$ . If it is fresh,  $VS_k$  sends  $ID_k$  and requests the sink node's master key  $X_k$  from  $RA$ . After receiving  $X_k$  from  $RA$  through a secure channel,  $VS_k$  computes  $M_{SV2}^* = h(ID_k||NS_{j2}||X_k||ID_j)$  and checks whether  $M_{SV2}^* \stackrel{?}{=} M_{SV2}$ . If it is verified,  $VS_k$  chooses a random nonce  $NV_{k2}$  and computes  $v = h(ID_k||NS_{j2}||NV_{k2})$ ,  $M_{VS2} = h(X_k||NS_{j2}||v)$ , and  $t = NS_{j2} \oplus NV_{k2}$ . Finally,  $VS_k$  sends  $\{M_{VS2}, t\}$  to  $SN_j$ .
- Step 5:** After receiving the message  $\{M_{VS2}, t\}$ ,  $SN_j$  retrieves  $NV_{k2} = t \oplus NS_{j2}$  and computes  $v = h(ID_k||NS_{j2}||NV_{k2})$  and  $M_{VS2}^* = h(X_k||NS_{j2}||v)$ . Then,  $SN_j$  checks whether  $M_{VS2}^* \stackrel{?}{=} M_{VS2}$  is correct. If it is correct,  $SN_j$  computes  $w = NS_{j2} \oplus NU_a$  and  $M_{ST2} = h(q_i||NU_a||NS_{j2}||ID_i||ID_k)$  and sends  $\{M_{ST2}, w\}$  to  $U_a$ .
- Step 6:** Upon receiving the message  $\{M_{ST2}, w\}$  from  $SN_j$ ,  $U_a$  successfully achieves mutual authentication.

Therefore, Mohit et al.'s scheme is vulnerable to impersonation attacks.

#### 4.2. Trace Attack and Anonymity Preservation

According to Section 4.1, an adversary  $U_a$  can obtain the real identities of the vehicle sensor and sink node. First,  $U_a$  retrieves the vehicle sensor's real identity  $ID_k = p_2 \oplus h(p_1||q_i)$  and then computes  $NS_j = d_1 \oplus h(ID_k)$ . Finally,  $U_a$  retrieves the sink node's real identity  $ID_j = d_2 \oplus ID_k$ . For this reason, Mohit et al.'s scheme does not prevent trace attack or provide anonymity.

#### 4.3. Mutual Authentication

In Section 4.1, we demonstrate that Mohit et al.'s scheme does not resist impersonation attacks. An adversary  $U_a$  can compute the login request message  $\{M_{TS}, p_1, p_2, E_i\}$  and successfully achieve mutual authentication with  $VS_k$ . In addition, the sink node  $SN_j$  cannot compute the authentication message  $M_{ST} = h(q_i||NU_i||NS_j||ID_i||ID_k)$  in the login and authentication phase because  $SN_j$  does not know the real identity of  $U_i$ . Therefore, Mohit et al.'s scheme does not provide secure mutual authentication.

#### 4.4. Session Key Security

Mohit et al. claimed that their scheme can provide session key security because an adversary cannot compute  $M_{TS} = h(q_i||B_i||NU_i)$ . However, we demonstrate that an adversary can compute the value  $M_{TS}$  in Section 4.1. Therefore, Mohit et al.'s scheme cannot achieve session key security.

### 5. Proposed Protocol

In this section, we propose a secure authentication protocol for WSNs in vehicle communications to resolve the security problems of Mohit et al.'s scheme [23]. Our proposed scheme consists of four phases: system setup, user registration, login and authentication and password change. In our protocol,



the system setup phase is equivalent to that of Mohit et al.’s scheme. The details of the other three phases are presented below.

5.1. User Registration Phase

When a new user  $U_i$  wants to first access the sink node as a traffic manager, he or she must first register with the sink node. The user registration phase of the proposed protocol is shown in Figure 5 and the detailed steps are as follows:

- Step 1:** The user  $U_i$  selects the identity  $ID_i$  and password  $PW_i$  and then generates a random number  $a_i$  to computes  $HPW_i = h(PW_i||a_i)$ . Then,  $U_i$  sends  $\{ID_i, HPW_i\}$  to the sink node  $SN_j$  via a secure channel.
- Step 2:** After receiving the registration request message from  $U_i$ ,  $SN_j$  generates a random unique identity  $TID_i$  for  $U_i$  and computes  $X_i = h(ID_i||K_S)$ ,  $A_i = X_i \oplus h(ID_i||HPW_i)$ ,  $B_i = h(HPW_i||X_i)$ , and  $C_i = X_i \oplus h(TID_i||K_S)$ . After that,  $SN_j$  stores  $\{A_i, B_i, TID_i\}$  in a smartcard, which it issues to  $U_i$  through a secure channel. Finally,  $SN_j$  stores  $\{TID_i, C_i\}$  in a database.
- Step 3:** Upon receiving the smartcard from  $SN_j$ ,  $U_i$  calculates  $Q_i = h(ID_i||PW_i) \oplus a_i$  and stores  $\{Q_i\}$  in the smartcard. Consequently,  $SN_j$  stores  $\{A_i, B_i, TID_i, Q_i\}$  in the smartcard.

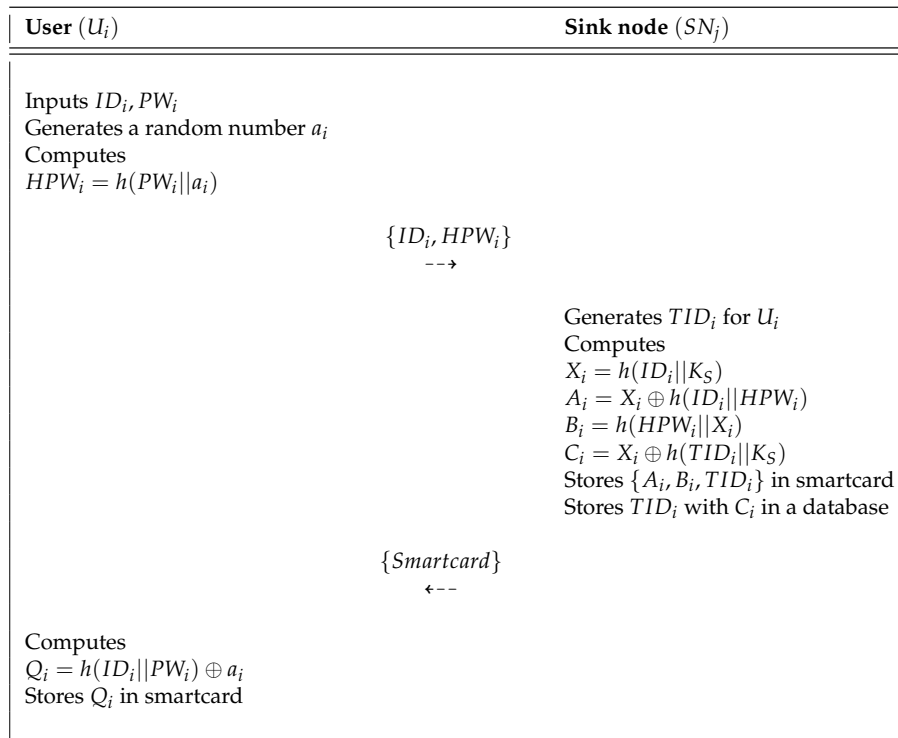


Figure 5. User registration phase of the proposed scheme.

5.2. Login and Authentication Phase

If a user  $U_i$  wants to access the sink node  $SN_j$ ,  $U_i$  must send a login request message. The login and authentication phase of our scheme is shown in Figure 6 and the details of this phase are as follows.

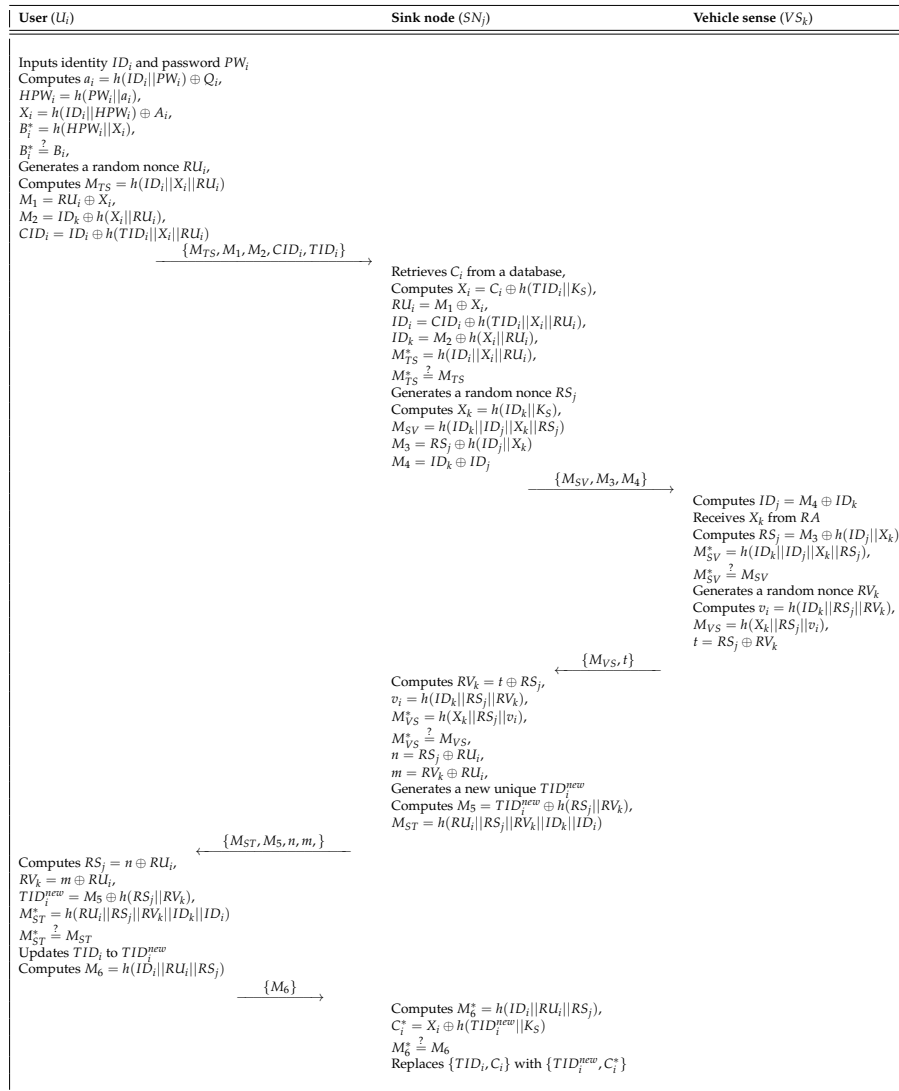


Figure 6. User login and authentication phase of the proposed scheme.

- Step 1:**  $U_i$  inserts the smartcard and inputs the identity  $ID_i$  and password  $PW_i$  into a smartcard reader. Then,  $U_i$  computes  $a_i = h(ID_i || PW_i) \oplus Q_i$ ,  $HPW_i = h(PW_i || a_i)$ ,  $X_i = h(ID_i || HPW_i) \oplus A_i$ , and  $B_i^* = h(HPW_i || X_i)$  and checks whether  $B_i^* \stackrel{?}{=} B_i$ . If it is equal,  $U_i$  generates a random nonce  $RU_i$  and computes  $M_1 = RU_i \oplus X_i$ ,  $M_2 = ID_k \oplus h(X_i || RU_i)$ ,  $CID_i = ID_i \oplus h(TID_i || X_i || RU_i)$ , and  $M_{TS} = h(ID_i || X_i || RU_i)$ .  $U_i$  sends the login request message  $\{M_{TS}, M_1, M_2, CID_i, TID_i\}$  to  $SN_j$  through a public channel.
- Step 2:** After receiving the login request message from  $U_i$ ,  $SN_j$  retrieves  $C_i$  matched with  $TID_i$  in a database. Then,  $SN_j$  computes  $X_i = C_i \oplus h(TID_i || K_S)$ ,  $RU_i = M_1 \oplus X_i$ ,  $ID_i = CID_i \oplus h(TID_i || X_i || RU_i)$ ,  $ID_k = M_2 \oplus h(X_i || RU_i)$ , and  $M_{TS}^* = h(ID_i || X_i || RU_i)$  and checks whether  $M_{TS}^* \stackrel{?}{=} M_{TS}$ . If it is correct,  $SN_j$  generates a random nonce  $RS_j$  and computes  $X_k = h(ID_k || K_S)$ ,  $M_{SV} = h(ID_k || ID_j || X_k || RS_j)$ ,  $M_3 = RS_j \oplus h(ID_j || X_k)$ , and  $M_4 = ID_k \oplus ID_j$ .  $SN_j$  also sends the authentication request message  $\{M_{SV}, M_3, M_4\}$  to  $VS_k$  via a public channel.
- Step 3:** Upon receiving the message  $\{M_{SV}, M_3, M_4\}$ ,  $VS_k$  computes  $ID_j = M_4 \oplus ID_k$  and receives  $X_k$  from RA. Then,  $VS_k$  computes  $RS_j = M_3 \oplus h(ID_j || X_k)$  and  $M_{SV}^* = h(ID_k || ID_j || X_k || RS_j)$  and checks whether  $M_{SV}^* \stackrel{?}{=} M_{SV}$ . If they are equal,  $VS_k$  generates a random nonce  $RV_k$  and computes  $v_i = h(ID_k || RS_j || RV_k)$ ,  $M_{VS} = h(X_k || RS_j || v_i)$ , and  $t = RS_j \oplus RV_k$ . Finally,  $VS_k$  sends  $\{M_{VS}, t\}$  to  $SN_j$  through a public channel.

- Step 4:** After receiving the message  $\{M_{VS}, t\}$  from  $VS_k$ ,  $SN_j$  computes  $RV_k = t \oplus RS_j, v_i = h(ID_k || RS_j || RV_k)$  and  $M_{VS}^* = h(X_k || RS_j || v_i)$ . Then,  $SN_j$  checks whether  $M_{VS}^* \stackrel{?}{=} M_{VS}$ . If it is equal,  $SN_j$  computes  $n = RS_j \oplus RU_i$  and  $m = RV_k \oplus RU_i$ . After that,  $SN_j$  generates a new random unique identity  $TID_i^{new}$  and computes  $M_5 = TID_i^{new} \oplus h(RS_j || RV_k)$  and  $M_{ST} = h(RU_i || RS_j || RV_k || ID_k || ID_i)$ .  $SN_j$  also sends the message  $\{M_{ST}, M_5, n, m\}$  to  $U_i$  via an open channel.
- Step 5:** Upon receiving the message  $\{M_{ST}, M_5, n, m\}$ ,  $U_i$  computes  $RS_j = n \oplus RU_i, RV_k = m \oplus RU_i, TID_i^{new} = M_5 \oplus h(RS_j || RV_k)$ , and  $M_{ST}^* = h(RU_i || RS_j || RV_k || ID_k || ID_i)$ . Then,  $U_i$  checks whether  $M_{ST}^* \stackrel{?}{=} M_{ST}$ . If it is equal,  $U_i$  updates  $TID_i$  to  $TID_i^{new}$ . Finally,  $U_i$  computes  $M_6 = h(ID_i || RU_i || RS_j)$  and sends the confirmation message  $\{M_6\}$  to  $SN_j$ .
- Step 6:** After receiving the message  $\{M_6\}$  from  $U_i$ ,  $SN_j$  computes  $M_6^* = h(ID_i || RU_i || RS_j)$  and  $C_i^* = X_i \oplus h(TID_i^{new} || K_S)$ . Then,  $SN_j$  checks whether  $M_6^* \stackrel{?}{=} M_6$ . If it is valid,  $SN_j$  replaces  $\{TID_i, C_i\}$  with  $\{TID_i^{new}, C_i^*\}$ .

5.3. Password Change Phase

In our proposed protocol,  $U_i$  can change the password when desired without the help of the sink node  $SN_j$ . The password change phase is shown in Figure 7 and the detailed steps of this phase are presented below:

- Step 1:**  $U_i$  inserts his or her smartcard into a card reader and inputs the identity  $ID_i$  and old password  $PW_i^*$ .
- Step 2:**  $SC$  computes  $a_i^* = h(ID_i^* || PW_i^*) \oplus Q_i, HPW_i^* = h(PW_i^* || a_i^*), X_i^* = h(ID_i^* || HPW_i^*) \oplus A_i$ , and  $B_i^* = h(HPW_i^* || X_i^*)$ . Then,  $SC$  compares the computed  $B_i^*$  with the stored  $B_i$  in its memory. If it is valid,  $SC$  sends an authentication message to  $U_i$ .
- Step 3:** On receiving the message from the smartcard,  $U_i$  inserts the new password  $PW_i^{new}$  in the smartcard.
- Step 4:** Using the new password  $PW_i^{new}$ ,  $SC$  computes  $Q_i^{new} = h(ID_i^* || PW_i^{new}) \oplus a_i^*, HPW_i^{new} = h(PW_i^{new} || a_i^*), A_i^{new} = X_i^* \oplus h(ID_i^* || HPW_i^{new}), B_i^{new} = h(HPW_i^{new} || X_i^*),$  and  $C_i^{new} = X_i^* \oplus h(TID_i || K_S)$ . Finally, the smartcard replaces the old information with  $\{A_i^{new}, B_i^{new}, C_i^{new}, Q_i^{new}\}$ .

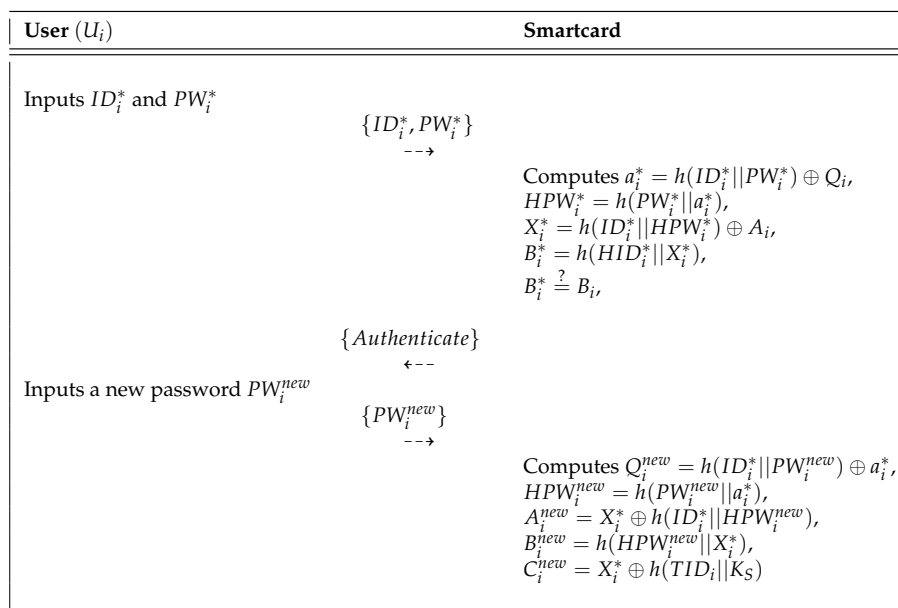


Figure 7. Password change phase of the proposed scheme.

## 6. Security Analysis

In this section, we use the Burrow–Abadi–Needham (BAN) logic [28], which is a broadly accepted formal security model, to carry out an analysis and prove that our protocol can provide secure mutual authentication. We also demonstrate that our proposed protocol can resist various attacks through an informal security analysis, which is based on Section 1.1.

### 6.1. Informal Security Analysis

We present an informal security analysis of our proposed scheme to show that it prevents trace, impersonation, and replay attacks. In addition, we demonstrate that our protocol can achieve mutual authentication and anonymity.

#### 6.1.1. Impersonation Attack

If an adversary  $U_a$  tries to impersonate a legitimate user  $U_i$ ,  $U_a$  must generate a login request message  $\{M_{TS}, M_1, M_2, CID_i, TID_i\}$  and response message  $\{M_6\}$  successfully. However,  $U_a$  cannot generate these because  $U_a$  cannot know the real identity of  $U_i$  and secret parameters  $X_i$ ,  $RU_i$ , and  $K_S$ . In addition,  $U_a$  does not retrieve a random nonce  $RU_i$  from  $M_1$ . Therefore, our protocol resists impersonation attacks because  $U_a$  cannot generate valid messages.

#### 6.1.2. Trace Attack and Anonymity

In the login and authentication phase of our protocol, an adversary  $U_a$  cannot trace a legitimate user  $U_i$  or vehicle  $VS_k$  because all transmitted messages are changed every session. In addition,  $U_i$  sends the dynamic identity  $CID_i = ID_i \oplus h(TID_i || X_i || RU_i)$  and  $TID_i$  to the sink node, and the identity of  $VS_k$  is also included in  $M_4 = ID_k \oplus ID_j$ . In other words, to obtain the record of a user's movement and real identity, an adversary must know the user's real identity  $ID_i$ , secret parameter  $X_i$ , and random nonces  $RU_i$ ,  $RS_j$ , and  $RV_k$ . For these reasons, our protocol provides the anonymity and is secure against trace attacks.

#### 6.1.3. Smartcard Stolen Attack

According to Section 1.1, we assume that an adversary  $U_a$  can obtain a smartcard and extract the parameters  $\{A_i, B_i, TID_i, Q_i\}$ . However,  $U_a$  cannot obtain any sensitive user information without  $ID_i$  and  $PW_i$  because the parameters stored in smartcards are masked in  $X_i = h(ID_i || K_S)$ ,  $A_i = X_i \oplus h(ID_i || HPW_i)$ ,  $B_i = h(HPW_i || X_i)$ ,  $C_i = X_i \oplus h(TID_i || K_S)$ , and  $Q_i = h(ID_i || PW_i) \oplus a_i$  by the hash function and XOR operation. Consequently, our proposed protocol prevents smartcard stolen attack.

#### 6.1.4. Replay Attack

According to Section 1.1, we suppose that adversary  $U_a$  tries to impersonate a legitimate user  $U_i$  by resending messages transmitted in the previous session,  $U_a$  cannot impersonate  $U_i$  successfully. In our scheme, the sink node  $SN_j$  checks whether a random nonce is fresh or not. If a random nonce value  $RU_i$  is not fresh,  $SN_j$  rejects the login request message. In addition,  $U_a$  cannot generate the confirmation message  $M_6$  successfully because  $U_a$  cannot obtain the random nonce  $RS_j$  generated by  $SN_j$ . Therefore, the proposed protocol is secure against replay attacks.

#### 6.1.5. Secure Mutual Authentication

When receiving the login message  $\{M_{TS}, M_1, M_2, CID_i, TID_i\}$  and confirmation message  $\{M_6\}$  from  $U_i$ , the sink node  $SN_j$  checks whether  $M_{TS}$  and  $M_6$  are correct. In addition,  $SN_j$  retrieves  $X_i$  from a database to validate  $M_{TS}$ . If this is correct,  $SN_j$  authenticates  $U_i$ . After receiving  $\{M_{VS}, t\}$  from  $VS_k$ , the sink node checks whether  $M_{SV} = h(ID_k || RS_j || RV_k)$  is valid. If it is valid,  $SN_j$  authenticates  $VS_k$ . Finally, the user  $U_i$  checks whether the received value  $M_{ST} = h(RU_i || RS_j || RV_k || ID_k || ID_i)$  is correct.

If it is correct,  $U_i$  authenticates  $SN_j$ . Therefore, all entities authenticate each other successfully because an adversary cannot know the important parameters discussed in Sections 6.1.1 and 6.1.2.

According to Sections 6.1.2 and 6.1.5, all transmitted messages are changed every session and an adversary cannot obtain user's sensitive information. Therefore, we achieve essential security requirement into untraceability, anonymity, secure mutual authentication and confidentiality. Furthermore, secure mutual authentication is proved in Section 6.2 using BAN logic.

## 6.2. Security Analysis Using BAN Logic

To prove the secure mutual authentication of our protocol, we perform an analysis with the BAN logic [28], which is a widely accepted formal security model. First, we define the notation of the BAN logic in Table 2. Then, we describe the logical postulates of the BAN logic in Section 6.2.1. Next, we present the goals, idealized form, and initial assumptions of our protocol. Finally, we demonstrate that our protocol achieves secure mutual authentication between  $U_i$  and  $VK_k$  by using the BAN logic.

**Table 2.** Notations of the BAN logic.

Notation	Description
$P \equiv X$	$P$ <b>believes</b> the statement $X$
$\#X$	The statement $X$ is <b>fresh</b>
$P \triangleleft X$	$P$ <b>sees</b> the statement $X$
$P \sim X$	$P$ once <b>said</b> $X$
$P \Rightarrow X$	$P$ <b>controls</b> the statement $X$
$\langle X \rangle_Y$	Formula $X$ is <b>combined</b> with the formula $Y$
$\{X\}_K$	Formula $X$ is <b>encrypted</b> by the key $K$
$P \stackrel{K}{\leftrightarrow} Q$	$P$ and $Q$ communicate using $K$ as the <b>shared key</b>
$SK$	Session key used in the current authentication session

### 6.2.1. Postulates of BAN Logic

The postulates of the BAN logic are given below:

1. Message meaning rule :

$$\frac{P \mid \equiv P \stackrel{K}{\leftrightarrow} Q, \quad P \triangleleft \{X\}_K}{P \mid \equiv Q \mid \sim X},$$

2. Nonce verification rule :

$$\frac{P \mid \equiv \#(X), \quad P \mid \equiv Q \mid \sim X}{P \mid \equiv Q \mid \equiv X},$$

3. Jurisdiction rule :

$$\frac{P \mid \equiv Q \mid \Rightarrow X, \quad P \mid \equiv Q \mid \equiv X}{P \mid \equiv X},$$

4. Freshness rule :

$$\frac{P \mid \equiv \#(X)}{P \mid \equiv \#(X, Y)},$$

5. Belief rule :

$$\frac{P \mid \equiv (X, Y)}{P \mid \equiv X}.$$

### 6.2.2. Goals

We have the following goals to prove the secure mutual authentication of our proposed protocol:

$$\text{Goal 1: } U_i \mid\equiv (RS_j, RV_k),$$

$$\text{Goal 2: } U_i \mid\equiv SN_j \mid\equiv (RS_j, RV_k),$$

$$\text{Goal 3: } SN_j \mid\equiv (RU_i),$$

$$\text{Goal 4: } SN_j \mid\equiv U_i \mid\equiv (RU_i),$$

$$\text{Goal 5: } SN_j \mid\equiv (RV_k),$$

$$\text{Goal 6: } SN_j \mid\equiv VS_k \mid\equiv (RV_k).$$

### 6.2.3. Idealized Forms

The idealized forms of the transmitted messages are given below:

$$\text{Msg1: } U_i \rightarrow SN_j: (ID_i, ID_k, TID_i, RU_i)_{X_i},$$

$$\text{Msg2: } SN_j \rightarrow VS_k: (ID_i, ID_k, RU_i)_{X_k},$$

$$\text{Msg3: } VS_k \rightarrow SN_j: (ID_k, RS_j, RV_k)_{X_k},$$

$$\text{Msg4: } SN_j \rightarrow U_i: (ID_k, TID_i^{new}, RU_i, RS_j, RV_k)_{ID_i},$$

$$\text{Msg5: } U_i \rightarrow SN_j: (RU_i, RS_j)_{ID_i}.$$

### 6.2.4. Assumptions

We make the following initial assumptions to perform the BAN logic proof:

$$A_1: U_i \mid\equiv (U_i \xleftrightarrow{X_i} SN_j),$$

$$A_2: SN_j \mid\equiv (U_i \xleftrightarrow{X_i} SN_j),$$

$$A_3: VS_k \mid\equiv (VS_k \xleftrightarrow{X_k} SN_j),$$

$$A_4: SN_j \mid\equiv (VS_k \xleftrightarrow{X_k} SN_j),$$

$$A_5: SN_j \mid\equiv \#(RU_i),$$

$$A_6: VS_k \mid\equiv \#(RS_j),$$

$$A_7: SN_j \mid\equiv \#(RV_k),$$

$$A_8: U_i \mid\equiv \#(RS_j),$$

$$A_9: U_i \mid\equiv (U_i \xleftrightarrow{ID_i} SN_j),$$

$$A_{10}: U_i \mid\equiv SN_j \Rightarrow (RS_j, RV_k),$$

$$A_{11}: SN_j \mid\equiv U_i \Rightarrow (RU_i),$$

$$A_{12}: SN_j \mid\equiv VS_k \Rightarrow (RV_k).$$

### 6.2.5. Proof Using BAN Logic

The detailed steps of the main proof are as follows:

**Step 1:** According to  $Msg_1$ , we can obtain

$$S_1 : SN_j \triangleleft (ID_i, ID_k, TID_i, RU_i)_{X_i}.$$

**Step 2:** In conformity with the message meaning rule with  $S_1$  and  $A_2$ , we can get

$$S_2 : SN_j \mid\equiv U_i \sim (ID_i, ID_k, TID_i, RU_i)_{X_i}.$$

**Step 3:** According to the freshness rule with  $A_5$ , we can get

$$S_3 : SN_j \mid\equiv \#(ID_i, ID_k, TID_i, RU_i)_{X_i}.$$

**Step 4:** According to the nonce verification rule with  $S_2$  and  $S_3$ , we can obtain

$$S_4 : SN_j \mid\equiv U_i \mid\equiv (ID_i, ID_k, TID_i, RU_i)_{X_i}.$$

**Step 5:** According to  $Msg_2$ , we can get

$$S_5 : VS_k \triangleleft (ID_i, ID_k, RU_i)_{X_k}.$$

**Step 6:** In conformity with the message meaning rule with  $S_5$  and  $A_3$ , we can get

$$S_6 : VS_k \mid\equiv SN_j \sim (ID_i, ID_k, RU_i)_{X_k}.$$

**Step 7:** According to the freshness rule with  $A_6$ , we can obtain

$$S_7 : VS_k \mid\equiv \#(ID_i, ID_k, RU_i)_{X_k}.$$

**Step 8:** According to the nonce verification rule with  $S_6$  and  $S_7$ , we can get

$$S_8 : VS_k \mid\equiv SN_j \mid\equiv (ID_i, ID_k, RU_i)_{X_k}.$$

**Step 9:** According to  $Msg_3$ , we can obtain

$$S_9 : SN_j \triangleleft (ID_k, RS_j, RV_k)_{X_k}.$$

**Step 10:** In conformity with the message meaning rule with  $S_9$  and  $A_4$ , we can obtain

$$S_{10} : SN_j \mid\equiv VS_k \sim (ID_k, RS_j, RV_k)_{X_k}.$$



**Step 11:** According to the freshness rule with  $A_7$ , we can get

$$S_{11} : SN_j \mid \equiv \#(ID_k, RS_j, RV_k)_{X_k}.$$

**Step 12:** According to the nonce verification rule with  $S_{10}$  and  $S_{11}$ , we can get

$$S_{12} : SN_j \mid \equiv VS_k \mid \equiv (ID_k, RS_j, RV_k)_{X_k}.$$

**Step 13:** According to  $Msg_4$ , we can obtain

$$S_{13} : U_i \triangleleft (ID_k, TID_i^{new}, RU_i, RS_j, RV_k)_{ID_i}.$$

**Step 14:** In conformity with the message meaning rule with  $S_{13}$  and  $A_9$ , we can get

$$S_{14} : U_i \mid \equiv SN_j \sim (ID_k, TID_i^{new}, RU_i, RS_j, RV_k)_{ID_i}.$$

**Step 15:** According to the freshness rule with  $A_8$ , we can get

$$S_{15} : U_i \mid \equiv \#(ID_k, TID_i^{new}, RU_i, RS_j, RV_k)_{ID_i}.$$

**Step 16:** According to the nonce verification rule with  $S_{14}$  and  $S_{15}$ , we can get

$$S_{16} : ID_i \mid \equiv SN_j \mid \equiv (ID_k, TID_i^{new}, RU_i, RS_j, RV_k)_{ID_i}.$$

**Step 17:** According to the belief rule with  $S_{16}$ , we can get

$$S_{17} : U_i \mid \equiv SN_j \mid \equiv (RS_j, RV_k). \quad \textbf{(Goal 2)}$$

**Step 18:** In conformity with the jurisdiction rule with  $S_{17}$  and  $A_{10}$ , we can obtain

$$S_{18} : U_i \mid \equiv (RS_j, RV_k). \quad \textbf{(Goal 1)}$$

**Step 19:** In conformity with the belief rule with  $S_4$ , we can get

$$S_{19} : SN_j \mid \equiv U_i \mid \equiv (RU_i). \quad \textbf{(Goal 4)}$$

**Step 20:** According the jurisdiction rule with  $S_{19}$  and  $A_{11}$ , we can obtain

$$S_{20} : SN_j \mid \equiv (RU_i). \quad \textbf{(Goal 3)}$$

**Step 21:** In conformity with the belief rule with  $S_{12}$ , we can get

$$S_{21} : SN_j \mid \equiv VS_k \mid \equiv (RV_k). \quad \textbf{(Goal 6)}$$

**Step 22:** According the jurisdiction rule with  $S_{19}$  and  $A_{11}$ , we can obtain

$$S_{20} : SN_j \mid \equiv (RV_k). \quad \textbf{(Goal 5)}$$

Based on goals 1–6, we prove that our proposed protocol achieves secure mutual authentication between  $U_i$  and  $VS_k$ .

## 7. Security Analysis Using the AVISPA Tool

In this section, we perform a formal security verification of our protocol with the widely accepted Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation tool [33,34]. Formal security verification with this tool has received much attention and has been used in numerous studies to demonstrate that various authentication protocols are secure against replay and man-in-the-middle attacks [35–39].

With AVISPA, the security protocol must be implemented by using the High Level Protocols Specification Language (HLPSL) [40]. The HLPSL specifications of the security protocol are translated to an intermediate format (IF) by the HLP2IF translator. Finally, it is converted to the output format (OF) with the On-the-fly Model-Checker (OFMC) [41], the CL-based Attack Searcher (AtSe) [42], SAT-based Model-Checker (SATMC), or Tree Automata-based Protocol Analyzer (TA4SP).

### 7.1. HLPSL Specifications

According to HLPSL, the proposed protocol has three entities, which are called *role*: *user* denotes a user  $UA$ , *sinknode* denotes a sink node  $SN$ , and *vehiclesense* denotes a vehicle sense  $VS$ . The *session* and *environment* also contain the security goals, as shown in Figure 8. The role specifications of  $U_i$  are shown in Figure 9 and the details are as follows.

```

role session (UA, SN, VS : agent ,
SKuasn : symmetric_key,
H: hash_func)
def=
local S1,S2,S3,R1,R2,R3:channel(dy)
composition
user(UA,SN,VS,SKuasn,H,S1,R1)
^sinknode(UA,SN,VS,SKuasn,H,S2,R2)
^vehiclesense(UA,SN,VS,H,S3,R3)
end role

```

(a) Session

```

role environment()
def=
const ua,sn,vs : agent,
skuasn : symmetric_key,
cidi,tidi,t,n,m: text,
h : hash_func,
ua_sn_rui,vs_sn_rvk,sn_vs_rsj : protocol_id,
sp1, sp2, sp3, sp4 : protocol_id
intruder_knowledge = {cidi,tidi,t,n,m,ua,sn,vs,h}
composition
session(ua,sn,vs,skuasn,h)
^session(i,sn,vs,skuasn,h)
^session(ua,i,vs,skuasn,h)
^session(ua,sn,i,skuasn,h)
end role
goal
secrecy_of s1
secrecy_of s2
secrecy_of s3
secrecy_of s4
authentication_on ua_sn_rui
authentication_on vs_sn_rvk
authentication_on sn_vs_rsj
end goal
environment()

```

(b) Environment

Figure 8. Role specification for session and environment.

```

%% role User
role user(UA, SN, VS :agent,
  SKuasn : symmetric_key,
  H: hash_func,
  SND, RCV : channel(dy))
played_by UA
def=
local State : nat,
  IDi,PWi,Idk,Idj,Aii,Ai, Bi, Ci, HPWi,Ks,TIDi,
  M1,M2,M3,M4,M5,M6,CIDi,Mt s,Msv, Mvs,Mst, T,
  RUi,RSj,RVk, Vi, Xk, TIDnew, N,M :text
const s1,s2,s3,s4,ua_sn_rui,vs_sn_rvk,sn_vs_rs_j : protocol_id
init State := 0
transition
%% Registration phase
1.State = 0 ∧ RCV(start) =>
State' := 1 ∧ Aii' := new() ∧ HPWi' := H(PWi.Aii')
% Send registration request <IDi, HPWi> to SN via secure channel
  ∧ SND({IDi,HPWi}_SKuasn)
  ∧ secret({IDi,HPWi},s1,{UA,SN})
  ∧ secret({PWi},s2,{UA})
% Receive smart card from SN via secure channel
2.State = 1 ∧ RCV({xor(H(IDi.Ks'),H(IDi.H(PWi.Aii'))),
  H(xor(H(PWi.Aii'),H(IDi.Ks'))),TIDi')_SKuasn) =>
%% Authentication phase
State' := 2 ∧ RUi' := new()
  ∧ M1' := xor(RUi',H(IDi.Ks'))
  ∧ M2' := xor(IDk,H(H(IDi.Ks'),RUi'))
  ∧ CIDi' := xor(IDi,H(TIDi',H(IDi.Ks'),RUi'))
  ∧ Mts' := H(IDi.H(IDi.Ks'),RUi')
% Send message <Mts,M1,M2,CIDi,TIDi> to SN via open channel
  ∧ SND(Mts'.M1',M2',CIDi',TIDi')
% UA has generated random nonce RUi for SN
  ∧ witness(UA,SN,ua_sn_rui, RUi')
% Receive message <Mst,M5,N,M> from SN via open channel
3.State = 2 ∧ RCV(H(RUj',RSj',RVk'),xor(RSj',RUj').xor(RVk',RUj'))=>
State' := 3 ∧ M6' := H(IDi.RUj',RSj')
% Send message <M6> to SN via open channel
  ∧ SND(M6')
end role

```

Figure 9. Role specification for user UA.

When  $U_i$  receives the start message, UA changes the state value 0 to 1. Then, UA sends the registration request  $\{ID_i, HPW_i\}$  to SN via a secure channel and receives the smartcard from SN. After that, UA updates the state from 1 to 2. During the login and authentication phase, UA sends the login message  $\{M_{ts}, M_1, M_2, CID_i, TID_i\}$  to SN via a public channel. Then, UA declares  $witness(UA, SN, ua\_sn\_rui, RU_i')$ , which means that it generates a random nonce  $RU_i$ . After generating  $RU_i$ , UA receives the message  $\{M_{st}, M_5, n, m\}$  from SN and updates the state from 2 to 3. Finally, UA sends  $\{M_6\}$  to SN through a public channel and SN authenticates UA by using a random nonce  $RU_i$ . Similarly, the simulated results of SN and VS are defined as shown in Figures 10 and 11.

```

%% role vehicle sense
role vehiclesense(UA, SN, VS :agent,
  H: hash_func,
  SND, RCV : channel(dy))
played_by VS
def=
local State : nat,
  IDi,PWi,Idk,Idj,Aii,Ai, Bi, Ci, HPWi,Ks,TIDi,
  M1,M2,M3,M4,M5,M6,CIDi,Mts,Msv, Mvs,Mst, T,
  RUi,RSj,RVk, Vi, Xk, TIDnew, N,M :text
const s1,s2,s3,s4,ua_sn_rui,vs_sn_rvk,sn_vs_rs_j : protocol_id
init State := 0
transition
%% Authentication phase
% Receive <Msv,M3,M4> from SN via open channel
1.State = 0 ∧ RCV(H(IDk.IDj.H(IDk.Ks'),RSj'),
  xor(RSj',H(IDj.H(IDk.Ks'))),xor(IDk,IDj)) =>
State' := 1 ∧ RVk' := new() ∧ Vi' := H(IDk.RSj',RVk')
  ∧ Mvs' := H(H(IDk.Ks'),RSj',Vi')
  ∧ T' := xor(RSj',RVk')
% Send <Mvs, T> to SN via open channel
  ∧ SND(Mvs',T')
% VS has generated random nonce RVk for SN
  ∧ witness(VS,SN,vs_sn_rvk,RVk')
% VS's acceptance of the value RSj for VS by SN
  ∧ request(SN,VS,sn_vs_rs_j,RSj')
end role

```

Figure 10. Role specification for VS.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% role sink node
role sinknode(UA, SN, VS : agent,
  SKuasn : symmetric_key,
  H: hash_func,
  SND, RCV : channel(dy))
played_by SN
def=
local State : nat,
  IDi,PWi,IDk,IDj,Aii,Xi,Ai, Bi, Ci, HPWi,Ks,TIDi,
  M1,M2,M3,M4,M5,M6,CIDi,Mts,Msv, Mvs,Mst, T,
  RUi,RSj,RVk, Vi, Xk, TIDnew, N,M :text
  const s1,s2,s3,s4,ua_sn_rui,vs_sn_rvk,sn_vs_rs_j : protocol_id
init State := 0
transition
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Registration phase
% Receive message < IDi,HPWi> from UA via secure channel
1. State = 0  $\wedge$  RCV({IDi,H(PWi,Aii)}_SKuasn) =>
State' := 1  $\wedge$  Ks' := new()  $\wedge$  TIDi' := new()
 $\wedge$  Xi' := H(IDi.Ks')
 $\wedge$  Ai' := xor(H(IDi.Ks'),H(IDi,H(PWi,Aii)))
 $\wedge$  Bi' := H(xor(H(PWi,Aii),Xi'))
 $\wedge$  Ci' := xor(Xi',H(TIDi.Ks'))
% Send message <Ai,Bi,TIDi> to UA via secure channel
 $\wedge$  SND({Ai'.Bi'.TIDi'}_SKuasn)
 $\wedge$  secret({IDi,HPWi},s1,{UA,SN})
 $\wedge$  secret({Ks'},s3,{SN})
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Authentication phase
% Receive message <Mts,M1,M2,CIDi,TIDi> from UA via open channel
2. State = 1  $\wedge$  RCV(H(IDi.H(IDi.Ks').RUi').xor(RUi',H(IDi.Ks')).
xor(IDk,H(IDi.Ks').RUi')).
xor(IDi,H(TIDi'.H(IDi.Ks').RUi')),TIDi') =>
State' := 2  $\wedge$  RSj' := new()
 $\wedge$  Xk' := H(IDk.Ks')  $\wedge$  Msv' := H(IDk.IDj.Xk'.RSj')
 $\wedge$  M3' := xor(RSj',H(IDj.Xk'))
 $\wedge$  M4' := xor(IDk.IDj)
% Send message <Msv,M3,M4> to VS via open channel
 $\wedge$  SND(Msv'.M3'.M4')
% SN has generated random nonce RSj for VS
 $\wedge$  witness(SN,VS,sn_vs_rs_j)
% Receive message <Mvs,t> for VS via open channel
3. State = 2  $\wedge$  RCV(H(H(IDk.Ks').RSj'.H(IDk.RSj'.RVk')).xor(RSj',RVk')) =>
State' := 3  $\wedge$  RUi' := new()  $\wedge$  N' := xor(RSj',RUi')  $\wedge$  M' := xor(RVk',RUi')
 $\wedge$  TIDnew' := new()
 $\wedge$  M5' := xor(TIDnew', H(RSj'.RVk'))
 $\wedge$  Mst' := H(RUi'.RSj'.RVk'.IDk.IDi)
% Send message <Mst,M5,N,M> to UA via open channel
 $\wedge$  SND(Mst'.M5'.N'.M')
% SN's acceptance of the value RVk for SN by VS
 $\wedge$  request(VS,SN,vs_sn_rvk,RVk')
 $\wedge$  secret({TIDnew'},s4,{UA,SN})
% Receive message <M6> from UA via open channel
4. State = 3  $\wedge$  RCV(H(IDi.RUi'.RSj')) =>
% SN's acceptance of the value RUi for SN by UA
State' := 4  $\wedge$  request(UA,SN,ua_sn_rui, RUi')

end role

```

Figure 11. Role specification for SN.

<pre> % OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/ysj.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 1.17s visitedNodes: 130 nodes depth: 6 plies </pre>	<pre> SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/ysj.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 2 states Reachable : 0 states Translation: 0.12 seconds Computation: 0.00 seconds </pre>
--	--

Figure 12. The result of analysis using OFMC and CL-AtSe

## 7.2. Analysis of Simulation Results

In this section, we present the results of the AVISPA analysis using OFMC and CL-AtSe back-ends to ensure the security of our protocol, as shown in Figure 12. To estimate the security against replay

attack, the OFMC and CL-AtSe back-ends check whether a legitimate entity can execute the protocol by searching for a passive adversary. Moreover, the OFMC and CL-AtSe back-ends also check whether the proposed protocol is secure against the man-in-the-middle attack for the DY model checking.

The OFMC back-end has a search time of 1.17 seconds to visit 130 nodes, and the CL-AtSe back-end analyzes two states with a translation time of 0.12 seconds. Because the replay attack and Dolev–Yao model checking are performed successfully, the proposed protocol is safe against replay and man-in-the-middle attacks.

## 8. Performance Analysis

In this section, we compare the computation and communication costs of our proposed protocol with those of related protocols [3,15,16,23,43,44] and discuss the security properties.

### 8.1. Computation Cost

We compare the computation overheads of our protocol with those of related protocols [3,15,16,23,43,44]. For the comparison of computation cost, we define the notations as follows.  $T_h$ ,  $T_s$ , and  $T_M$  denote the times for hash operation ( $\approx 0.0005$  s), symmetric key cryptographic operation ( $\approx 0.0087$  s) and elliptic curve scalar point multiplication operation ( $\approx 0.0630$  s), respectively. The analysis results are presented in Table 3.

**Table 3.** Computation cost of our proposed scheme with other related schemes.

Schemes	User	Sink Node	Sensor	Total Cost	Total Cost (s)
Shi et al. [15]	$5T_h + 3T_M$	$3T_h + 2T_M$	$4T_h + T_M$	$12T_h + 6T_M$	0.3840
Choi et al. [16]	$12T_h + 3T_M$	$5T_h + T_M$	$7T_h + 2T_M$	$24T_h + 6T_M$	0.3900
He et al. [43]	$4T_h + 2T_s$	$2T_h + 5T_s$	$T_h + 2T_s$	$7T_h + 9T_s$	0.0818
Xue et al. [44]	$10T_h$	$14T_h$	$6T_h$	$30T_h$	0.0150
Kumari and Om [3]	$10T_h$	$8T_h$	$6T_h$	$24T_h$	0.0120
Mohit et al. [23]	$7T_h$	$9T_h$	$4T_h$	$20T_h$	0.0100
Ours	$8T_h$	$13T_h$	$4T_h$	$25T_h$	0.0125

$T_h$ : One-way hash operation,  $T_s$ : Symmetric key cryptographic operation,  $T_M$ : Elliptic curve scalar point multiplication operation.

We use the existing computation analysis results of Mohit et al. [23] for a rough evaluation. We do not include the XOR operation because it is negligible compared with the other operations. The results show that our protocol needs  $8T_h$  for the user,  $13T_h$  for the sink node, and  $4T_h$  for the sensor. Thus, total cost of our protocol is 0.0125 seconds. Even though this is slightly higher than the cost for Mohit et al.'s protocol, the difference is negligible, and the proposed protocol provides better security than other protocols. Therefore, our protocol is secure and suitable for practical WSNs environments.

### 8.2. Security Properties

Table 4 compares the security properties of our proposed protocol compared with other related protocols. The existing related schemes clearly cannot resist various attacks, and their protocols cannot achieve anonymity and mutual authentication. For these reasons, our protocol provides better security features than the other protocols [3,15,16,23,43,44].

**Table 4.** Security properties of our proposed scheme with other related schemes.

Security Property	Shi et al. [15]	Choi et al. [16]	He et al. [43]	Xue et al. [44]	Kumari and Om [3]	Mohit et al. [23]	Ours
Impersonation attack	○	○	○	○	×	×	○
Smartcard stolen attack	×	○	○	○	○	×	○
Password change attack	○	×	×	×	○	○	○
Replay attack	○	○	○	○	○	○	○
Trace attack	×	×	×	×	×	×	○
Anonymity	×	×	○	×	×	×	○
Mutual authentication	○	○	○	○	×	×	○

○: preserves the security properties, ×: does not preserve the security properties.

### 8.3. Communication Cost

Finally, we analyze the communication cost of our scheme with related protocols. For the communication analysis, we assume that a random nonce (number) and timestamp are 64 bits, a pseudo-identity is 160 bits, the SHA-1 hash digest [45] is 160 bits, elliptic curve scalar multiplication is 512 bits, and symmetric key cryptographic operation is 256 bits. In the login and authentication phase of our protocol, the transmitted messages  $\{M_{TS}, M_1, M_2, CID_i, TID_i\}$ ,  $\{M_{SV}, M_3, M_4\}$ ,  $\{M_{VS}, t\}$ ,  $\{M_{ST}, M_5, n, m, \}$ , and  $\{M_6\}$  require  $(160 + 64 + 64 + 160 + 160 = 608$  bits),  $(160 + 64 + 64 = 288$  bits),  $(160 + 64 = 224$  bits),  $(160 + 160 + 64 + 64 = 448$  bits) and 160 bits, respectively. Consequently, the total communication cost is  $(608 + 288 + 224 + 448 + 160 = 1728$  bits). Table 5 presents the results of this analysis. Even though our protocol has a higher communication cost than Mohit et al.'s scheme, the vehicle sense sends only 224 bits, which is similar to that of their scheme. Therefore, from the perspective of limited resources, the proposed scheme is sufficiently applicable to WSN environments.

**Table 5.** Communication cost of our proposed scheme with other related schemes.

Schemes	Communication Cost
Shi et al. [15]	3968 bits
Choi et al. [16]	3584 bits
He et al. [43]	1216 bits
Xue et al. [44]	1920 bits
Kumari and Om [3]	2048 bits
Mohit et al. [23]	1280 bits
Ours	1728 bits

## 9. Conclusions

In this paper, we demonstrate that Mohit et al.'s scheme does not resist the impersonation and trace attacks. We also show that it does not achieve secure mutual authentication, session key security, and anonymity. We propose a secure authentication protocol for WSNs in vehicular communications to resolve the security problems of their scheme. The proposed protocol is secure against impersonation, replay, smartcard stolen and trace attacks and can achieve secure mutual authentication and anonymity by using dynamic values for the transmitted messages that change every session. We also prove that our protocol can provide secure mutual authentication between  $U_i$ ,  $SN_j$  and  $VS_k$  by using BAN logic and we present a formal security verification using the AVISPA tool. Furthermore, we compare the performance and security functionalities with those of other related protocols. Therefore, the proposed protocol can be efficiently applied to practical vehicle communications systems.

**Author Contributions:** Conceptualization, S.Y.; Formal Analysis, K.P.; Project Administration, Y.P.; Software, J.L.; Supervision, Y.P.; Writing—Original Draft, S.Y.; Writing—Review and Editing, K.L., K.P. and Y.P.

**Funding:** This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning under Grant 2017R1A2B1002147 and in part by the BK21 Plus project funded by the Ministry of Education, Korea under Grant 21A20131600011.

**Acknowledgments:** The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback on the paper, which helped us to improve its quality and presentation.

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

## References

1. Chatterjee, K.; De, A.; Gupta, D. A secure and efficient authentication protocol in wireless sensor network. *Wirel. Pers. Commun.* **2015**, *81*, 17–37. [[CrossRef](#)]
2. Kim, J.; Lee, D.; Jeon, D.; Lee, Y.; Won, D. Security analysis and improvements two-factor mutual authentication with key agreement in wireless sensor networks. *Sensors* **2014**, *14*, 6443–6462. [[CrossRef](#)] [[PubMed](#)]
3. Kumari, S.; Om, H. Authentication protocol for wireless sensor networks applications like safety monitoring in coal mines. *Comput. Netw.* **2016**, *104*, 137–154. [[CrossRef](#)]
4. Wang, D.; Wang, P. On the anonymity of two-factor authentication schemes for wireless sensor networks. *Comput. Netw.* **2014**, *73*, 41–57. [[CrossRef](#)]
5. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123. [[CrossRef](#)] [[PubMed](#)]
6. Jiang, Q.; MA, P.F.; Lu, X.; Tian, Y.L. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2015**, *8*, 1070–1081. [[CrossRef](#)]
7. Amin, R.; Biswas, G.P. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.* **2016**, *36*, 58–80. [[CrossRef](#)]
8. Amin, R.; Hafizul Islam, S.K.; Biswas, G.P.; Khan M.K.; Leng, L.; Kumar, N. Design of an anonymity preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* **2016**, *101*, 42–62. [[CrossRef](#)]
9. Wong, K.H.; Zheng, Y.; Cao, J.; Wang, S. A dynamic user authentication scheme for wireless sensor networks. In Proceedings of the 2006 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06), Taichung, Taiwan, 5–7 June 2006; Volume 1, pp. 1–8.
10. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090. [[CrossRef](#)]
11. Chen, T.H.; Shih, W.K. A robust mutual authentication protocol for wireless sensor networks. *ETRI J.* **2010**, *32*, 704–712. [[CrossRef](#)]
12. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. *Sensors* **2010**, *10*, 2450–2459. [[CrossRef](#)] [[PubMed](#)]
13. Yeh, H.L.; Chen, T.H.; Liu, P.C.; Kim, T.H.; Wei, H.W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2011**, *11*, 4767–4779. [[CrossRef](#)] [[PubMed](#)]
14. Han, W. Weakness of a Secured Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography. *IACR Cryptol. ePrint Arch.* **2011**, *2011*, 293.
15. Shi, W.; Gong, P. A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Int. J. Sens. Netw.* **2013**, *2013*, 730831. [[CrossRef](#)]
16. Choi, Y.; Lee, D.; Kim, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2014**, *14*, 10081–10106. [[CrossRef](#)] [[PubMed](#)]
17. Zhang, C.; Lin, X.; Lu, R.; Ho, P.H. RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks. In Proceedings of the 2008 IEEE International Conference on Communications, Beijing, China, 19–23 May 2008; pp. 1–7.
18. Zhang, C.; Lin, X.; Lu, R.; Ho, P.H.; Shen, S. An Efficient Message Authentication Scheme for Vehicular Communications. *IEEE Trans. Veh. Technol.* **2008**, *57*, 3357–3368. [[CrossRef](#)]
19. Lu, R.; Lin, X.; Zhu, H.; Ho, P.H.; Shen, X. ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In Proceedings of the 2008 IEEE INFOCOM Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 1–9.
20. Huang, D.; Misra, S.; Verma, M.; Xue, G. PACP: An efficient pseudonymous authentication-based conditional privacy protocol for VANETs. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 736–746. [[CrossRef](#)]
21. Chuang, M.C.; Lee, J.F. Team: Trust-extended authentication mechanism for vehicular ad hoc networks. *IEEE Syst. J.* **2014**, *8*, 749–758. [[CrossRef](#)]



22. Kumari, S.; Karuppiah, M.; Li, X.; Wu, F.; Das, A.K.; Odelu, V. An enhanced and secure trust-extended authentication mechanism for vehicular ad-hoc networks. *Secur. Commun. Netw.* **2016**, *9*, 4255–4271. [CrossRef]
23. Mohit, P.; Amin, R.; Biswas, G. Design of authentication protocol for wireless sensor network-based smart vehicular system. *Veh. Commun.* **2017**, *9*, 64–71. [CrossRef]
24. Alshaer, H.; Elmighani, J.M. Road safety based on efficient vehicular broadcast communications. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xian, China, 3–5 June 2009; pp. 1155–1160.
25. Alshaer, H. Securing vehicular ad-hoc networks connectivity with roadside units support. In Proceedings of the 2015 IEEE 8th GCC Conference & Exhibition, Muscat, Oman, 1–4 February 2015; pp. 1–6.
26. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
27. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology*; Springer Science+Business Media: Berlin, Germany; New York, NY, USA, 1999; pp. 388–397.
28. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
29. Zhang, L.; Wu, Q.; Domingo-Ferrer, J.; Qin, B.; Hu, C. Distributed Aggregate Privacy-Preserving Authentication in VANETs. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 516–526. [CrossRef]
30. Zhang, L.; Wu, Q.; Solanas, A.; Domingo-Ferrer, J. A Scalable Robust Authentication Protocol for Secure Vehicular Communications. *IEEE Trans. Veh. Technol.* **2009**, *59*, 1606–1617. [CrossRef]
31. Liu, J.; Li, J.; Zhang, L.; Dai, F.; Zhang, Y.; Meng, X.; Shen, J. Secure intelligent traffic light control using fog computing. *Future Gener. Comput. Syst.* **2018**, *78*, 817–824. [CrossRef]
32. Riley, M.; Akkaya, K.; Fong, K. A survey of authentication schemes for vehicular ad hoc networks. *Secur. Commun. Netw.* **2011**, *4*, 1137–1152. [CrossRef]
33. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: <http://www.avispa-project.org/> (accessed on 4 July 2018).
34. SPAN: A Security Protocol Animator for AVISPA. Available online: <http://www.avispa-project.org/> (accessed on 4 July 2018).
35. Park, K.S.; Park, Y.H.; Park, Y.H.; Reddy, A.G.; Das, A.K. Provably secure and efficient authentication protocol for roaming service in global mobility networks. *IEEE Access* **2017**, *5*, 25110–25125. [CrossRef]
36. Odelu, V.; Das, A.K.; Choo, K.R.; Kumar, N.; Park, Y.H. Efficient and secure time-key based single sign-on authentication for mobile devices. *IEEE Access* **2017**, *5*, 27707–27721. [CrossRef]
37. Odelu, V.; Das, A.K.; Kumari, S.; Huang, X.; Wazid, M. Provably secure authenticated key agreement scheme for distributed mobile cloud computing services. *Future Generat. Comput. Syst.* **2017**, *68*, 74–88. [CrossRef]
38. Park, K.S.; Park, Y.H.; Park, Y.H.; Das, A.K. 2PAKEP: Provably Secure and Efficient Two-Party Authenticated Key Exchange Protocol for Mobile Environment. *IEEE Access* **2018**, *6*, 30225–30241. [CrossRef]
39. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Kumar, N.; Park, Y.H.; Tanwar, S. Design of an Anonymity-Preserving Group Formation Based Authentication Protocol in Global Mobility Networks. *IEEE Access* **2018**, *6*, 20673–20693. [CrossRef]
40. Von Oheimb, D. The high-level protocol specification language HLPSSL developed in the EU project avispa. In Proceedings of the APPSEM 2005 Workshop, Tallinn, Finland, 13–15 September 2005; pp. 1–2.
41. Basin, D.; Modersheim, S.; Vigano, L. OFMC: A symbolic model checker for security protocols. *Int. J. Inf. Secur.* **2005**, *4*, 181–208. [CrossRef]
42. Turuani, M. The CL-Atse protocol analyser. In Proceedings of the International Conference on Rewriting Techniques and Applications (RTA), Seattle, WA, USA, 12–14 August 2006; pp. 227–286.
43. He, D.; Kumar, N.; Chen, J.; Lee, C.C.; Chilamkurti, N.; Yeo, S.S. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimed. Syst.* **2015**, *21*, 49–60. [CrossRef]
44. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal credential based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323. [CrossRef]
45. FIPS PUB 180-4: Secure Hash Standard (SHS). Available online: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf> (accessed on 23 July 2018).

