*electronics*

*Article*

# On the Security of a Secure and Computationally Efficient Authentication and Key Agreement Scheme for Internet of Vehicles

**Kisung Park [1], Myeonghyun Kim [2] and Youngho Park [3,*]**

[1] Department of Computer Engineering (Smart Security), Gachon University, Seongnam 13120, Republic of Korea; kisung@gachon.ac.kr
[2] System Security Research Section, Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea; kimmh12@etri.re.kr
[3] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea
* Correspondence: parkyh@knu.ac.kr

**Abstract:** In the Internet of Vehicles (IoV) environments, vehicles and roadside units (RSUs) communicate predominantly through public channels. These vehicles and RSUs exchange various data, such as traffic density, location, speed, etc. Therefore, secure and efficient authentication and key establishment (AKE) are needed to guarantee user privacy when exchanging data between vehicles and RSUs. Recently, a secure and computationally AKE scheme have been proposed to construct secure IoV environments. In their research, the authors asserted that their AKE scheme provides comprehensive security properties, protecting against various potential threats while simultaneously ensuring session key integrity, robust mutual authentication. This paper proved that the previous scheme does not prevent various attacks using logical and mathematical analyses. Moreover, we demonstrated that this scheme does not meet the essential security requirements and correctness of security assumptions. We perform the simulation proof using AVISPA, which is well known as a formal verification tool. To enhance the resilience of attacks, we propose solutions aimed at developing more robust and efficient AKE for IoV environments.

**Keywords:** authentication; Internet of Vehicles (IoV); session key security; security flaws

check for updates

**Citation:** Park, K.; Kim, M.; Park, Y. On the Security of a Secure and Computationally Efficient Authentication and Key Agreement Scheme for Internet of Vehicles. *Electronics* **2024**, *13*, 3136. https://doi.org/10.3390/electronics13163136

Academic Editor: Djuradj Budimir

Received: 16 July 2024
Revised: 1 August 2024
Accepted: 6 August 2024
Published: 8 August 2024

**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

The Internet of Vehicles (IoV) plays a crucial role in advancing intelligent transportation systems, significantly enhancing traffic efficiency and the capabilities of autonomous vehicles. The IoV architecture comprises a network of roadside units (RSUs), a central trusted authority (TA), and vehicles equipped with onboard units (OBUs). This interconnected system utilizes the "Dedicated Short Range Communications" (DSRCs) protocol for vehicle-to-RSU wireless communication, while RSUs maintain wired connections to the TA. By facilitating real-time exchange of critical data such as vehicle telemetry and traffic conditions, the IoV environments promise to create a more responsive and efficient transportation ecosystem.

Despite its promising potential, the IoV confronts significant challenges in data security and environmental sustainability [1]. The proliferation of sensitive information within the IoV system amplifies vulnerability to data leakages and potential attacks, necessitating robust security protocols [2,3]. The expanding vehicle population also contributes to escalating carbon emissions, demanding the integration of eco-friendly strategies. Therefore, to enhance security and minimize energy consumption, there is a critical need to develop a computationally efficient and sustainable AKE scheme for the IoV.

The contemporary AKE protocols in IoV systems typically employ a centralized architecture, wherein all vehicles authenticate only with the trusted authority (TA), while

roadside units (RSUs) function as intermediary nodes enabling vehicle–TA communication. The IoV environment is characterized by high-speed vehicular mobility, resulting in a highly dynamic network topology and communication patterns. Consequently, vehicles need to establish secure mutual AKE with the TA and exchange encrypted data within strict time limits. Moreover, many current AKE methods are based on computationally intensive public key cryptosystem as Elliptic Curve Cryptography (ECC). As traffic density rises, these centralized AKE with higher computational demands are more prone to encountering communication or computational bottlenecks at the TA. In high-density traffic environments, the timely transmission of messages between vehicles and the trusted authority (TA) becomes increasingly challenging. This situation could lead to extended authentication queues or necessitates multiple re-transmissions of authentication messages, resulting in substantial energy waste.

To overcome these issues, several research works have been proposed for ensuring efficiency and sustainability in IoV [4–10]. He et al. [4] introduced a multi-server AKE scheme that operates without the need for a trusted online third party. Ying and Nayak [5] presented an anonymous and lightweight AKE scheme utilizing smart cards. Their protocol significantly lowers communication and computational costs and is resistant to offline password guessing attacks. However, Chen et al. [6] proved that Ying and Nayak's [5] scheme does not resist several attacks such as fixation, replay, and offline identity guessing attacks. Chen et al. [6] also proposed a secure AKE scheme for IoV, which can reduce a computation cost during the authentication phases. However, Chen et al.'s [6] scheme suffers from high storage costs as it necessitates storing a large amount of data in memory [7]. Cui et al. [8] proposed a double pseudonym-based AKE scheme for IoV. Dua et al. [9] also proposed a robust AKE scheme for IoV, which has considered only vehicle-to-vehicle communications. Li et al. [10] proposed a privacy-preserving AKE scheme for data privacy in IoV. However, all the aforementioned works [6–10] have high computational cost and are unsuitable for workable IoV environments.

Recently, Xu et al. [11] presented an efficient AKE scheme for providing high security and computational efficiency in IoV environments. However, we demonstrate that Xu et al.'s scheme does not resist various security attacks, such as physical capture and impersonation. Since the security parameters stored in memory are easily leaked to an adversary, the adversary can obtain the secret parameters via a threat model.

To address these issues, research is being conducted on physically secure AKE schemes in vehicular network environments [12,13]. In 2024, Yu and Park [12] proposed the robust and anonymous AKE scheme using the physical unclonable function (PUF) for V2G networks. Wang et al. [13] also proposed a lightweight and physically secure AKE scheme for vehicular networks. In this paper, we propose solutions that use lightweight and physically secure techniques to enhance data security and protect user privacy.

This paper is organized as follows: Sections 1–3 present a review and cryptanalysis of Xu et al.'s [11] scheme. Section 4 offers a solution to enhance security and protection against attacks. Section 5 presents conclusions and future works.

### *1.1. Motivations and Contributions*

The main goal of our work is to prove the important security flaws of the Xu et al.'s scheme [11]. In their scheme, an attacker can easily generate the session key between entities using impersonation attacks. Therefore, we pointed out that Xu et al.'s scheme does not guarantee the security of a session key using formal (mathematical) security analysis, called the "Real-or-Random (ROR) model" [14], which is well known for formal security proof models [15,16]. Finally, we prove that their scheme is unsuitable for practical IoV environments and propose solutions for overcoming these flaws and enhancing its security.

### *1.2. Threat Model*

In Xu et al.'s scheme [11], they perform the security analysis under the "Dolev–Yao (DY) threat model" [17]. According to the DY threat model, an adversary can inject, delete,

modify and obtain the all public messages between communicating parties, including the vehicle, RSUs, and TA. The TA is a trusted entity that cannot be compromised by an adversary.

The adversary is unable to access the data stored in the RSU's memory because RSUs are equipped with a tamper-proof device (TPD) designed to prevent external data access. However, the adversary can freely capture and retrieve data from any number of OBUs. Through this capturing, the adversary can obtain all the secret information stored in the OBU's memory, as OBUs in vehicles lack physical protection [11].

## 2. Review of Xu et al.'s Scheme

This section concisely reviews Xu et al.'s [11] scheme and introduces the threat model for performing a cryptanalysis of their scheme. Xu et al.'s scheme consists of three phases: initialization, registration, and authentication. Table 1 shows the notations utilized in this paper.

**Table 1.** Notations used in this paper.

| Notation | Description |
|---|---|
| SA | A system administrator |
| RSU | A roadside unit |
| TA | A trusted authority |
| $K_{TA}$ | Master secret key of TA |
| $ID_{V/R}$ | Real identity of vehicle and RSU, respectively |
| $TID_{V/R}$ | Temporary identity of vehicle and RSU, respectively |
| $A_V, B_R, X_V$ | Authentication parameters |
| $P_{K_s}$ | The hash value of the previous session key |
| $r, n_i$ | Short-lived secret parameters |
| $K_S$ | Session key among the entities |
| $\oplus$ | XOR operation |
| $h()$ | The collision resistant hash function, $h : \{1,0\}^* \rightarrow Z_p^*$ |
| $\|\|$ | A concatenation |

### 2.1. Initialization Phase

This phase is carried out by TA to build up the initial parameters. The TA generates the master key $K_{TA}$ and stores it in memory of TA. The TA also selects the collision-resistant hash function $h$.

### 2.2. Registration Phase

In this phase, vehicles register with the TA to access the IoV networks. The detailed steps of the process are outlined below.

(1) The TA selects an identity $ID_{V/R}$ and a temporary identity $TID_{V/R}$ for vehicles and RSUs, respectively, and then TA chooses a nonce $P_{K_s}$ for each vehicle.
(2) The TA chooses a random number $r$, and computes $A_V = r \oplus K_{TA}$, $B_R = h(ID_R, K_{TA})$, $X_V = ID_V \oplus h(r, K_{TA})$. After that, the TA stores $\{ID_V, TID_V, A_V, P_{K_s}\}$ and $\{ID_R, TID_R, B_R, K_{TA}\}$ in the memory of the vehicle and RSU, respectively.
(3) Finally, the TA stores $\{ID_V, TID_V, A_V, P_{K_s}\}$ and $\{ID_R, TID_R, B_R, K_{TA}\}$ in the database.

### 2.3. Authentication and Key Agreement Phase

In this phase, the vehicle, TA, and RSU perform a mutual authentication to establish the session key. This phase is shown in Figure 1 and its detailed description is presented below.

| **Vehicle** ($VH_i$) | **Roadside Unit** (RSU) | **Trusted Authority** (TA) |
|---|---|---|
| Generates $n_1$, timestamp $t_1$<br>Computes $B_{V_i} = h(ID_V, P_{K_s})$,<br>$S_1 = n_1 \oplus B_V$<br>$S_2 = h(ID_V, TID_V, A_V, S_1, t_1, n_1)$<br>$\{TID_V, A_V, S_1, S_2, t_1\}$ $\longrightarrow$ | | |
| | Checks validity of $t_1$<br>Selects $n_2$, timestamp $t_2$<br>Computes $S_3 = n_2 \oplus B_R$,<br>$S_4 = h(TID_V, TID_R, ID_R, S_3, t_2, n_2)$<br>$\{(TID_V, TID_R, S_3, S_4, t_2)\}$ $\longrightarrow$ | |
| | | Checks validity of $t_2$<br>Checks if $TID_V, TID_R$ are in database.<br>Retrieves $\{TID_V, X_V, P_{K_s}\}$,<br>$\{TID_R, TID_R, B_R\}$, respectively.<br>Checks $S_4^* \stackrel{?}{=} S_4$<br>Generates nonce $n_3$, timestamp $t_3$<br>Computes $M_1 = h(n_2^*, n_3, K_{TA})$,<br>$S_5 = n_3 \oplus B_R$, $S_6 = M_1 \oplus P_{K_s}$,<br>$S_7 = h(ID_R, S_5, S_6, X_V, n_3, t_3)$<br>$\{S_5, S_6, S_7, X_V, t_3\}$ $\longleftarrow$ |
| | Checks validity of $t_3$<br>Computes $n_3^* = S_5 \oplus B_R$,<br>$S_7^* = h(ID_R, S_5, S_6, X_V, n_3^*, t_3)$,<br>Checks $S_7^* \stackrel{?}{=} S_7$<br>Computes $M_1 = h(n_2, n_3^*, K_{TA})$,<br>$P_{K_s} = M_1, \oplus S_6$, $r^* = A_V \oplus K_{TA}$,<br>$ID_V^* = X_V \oplus h(r^*, K_{TA})$,<br>$B_V = h(ID_V^*, P_{K_s})$, $n_1^* = S_1 \oplus P_{K_s}$,<br>$S_2^* = h(ID_V^*, TID_V, A_V, S_1, t_1, n_1^*)$ | |
| | Checks $S_2^* \stackrel{?}{=} S_2$<br>Generates $n_4$, $r^+$ and timestamp $t_4$,<br>Computes $K_s = h(n_1^*, n_4, P_{K_s})$,<br>$P_{K_s}^+ = h(n_1^*, n_4, K_s)$,<br>$X_V^+ = ID_V^* \oplus h(r^+, K_{TA})$,<br>$S_8 = n_2 \oplus M_1 \oplus P_{K_s}^+$,<br>$S_9 = n_3^* \oplus M_1, \oplus X_V^+$<br>$S_{10} = h(S_8, S_9, K_{TA}, n_2, n_3^*, t_4)$<br>$\{S_8, S_9, S_{10}, t_4\}$ $\longrightarrow$ | Checks validity of $t_4$<br>Computes $S_{10}^* = h(S_8, S_9, K_{TA}, n_2^*, n_3, t_4)$<br>Checks $S_{10}^* \stackrel{?}{=} S_{10}$<br>Computes $P_{K_s}^+ = n_2^* \oplus M_1 \oplus S_8$,<br>$X_V^+ = n_3 \oplus M_1 \oplus S_9$<br>Selects $TID_V^+$, $TID_R^+$, timestamp $t_5$<br>Calculates $M_2 = h(n_2^*, n_3, P_{K_s})$,<br>$M_3 = h(ID_R, n_2^*, n_3)$, $S_{11} = M_2 \oplus TID_V^+$,<br>$S_{12} = M_3 \oplus TID_R^+$,<br>$S_{13} = h(S_{11}, S_{12}, K_{TA}, M_2, M_3, t_5)$,<br>replaces $\{TID_V, X_V, P_{K_s}\}$ and $\{TID_R, ID_R, B_R\}$<br>with $\{TID_V, X_V, P_{K_s}, TID_V^+, X_V^+, P_{K_s}^+\}$<br>and $\{TID_R, ID_R, B_R, TID_R^+\}$, respectively.<br>$\{S_{11}, S_{12}, S_{13}, t_5\}$ $\longleftarrow$ |
| | Checks validity of $t_5$<br>Computes $M_2^* = h(n_2, n_3^*, P_{K_s})$,<br>$M_3^* = h(ID_R, n_2, n_3^*)$, | |
| Checks validity of $t_6$ | $S_{13}^* = h(S_{11}, S_{12}, K_{TA}, M_2^*, M_3^*, t_5)$ | |
| Computes $n_{4a}^* = S_{16} \oplus B_V$, | Checks $S_{13}^* \stackrel{?}{=} S_{13}$ | |
| $S_{17}^* = h(S_{14}, S_{15}, S_{16}, ID_V, n_4^*, t_6)$ | Generates timestamp $t_6$ | |
| Checks $S_{17}^* \stackrel{?}{=} S_{17}$ | Computes $TID_V^+ = M_2^* \oplus S_{11}$, | |
| Computes $M_4 = h(n_1, n_{4a}^*, ID_V)$, | $TID_R^+ = M_3^* \oplus S_{12}$, $A_V^+ = r^+ \oplus K_{TA}$, | |
| $A_{Va}^+ = M_4 \oplus S_{14}$, | $M_4 = h(n_1^*, n_4, ID_V^*)$, | |
| $TID_{Va}^+ = M_4 \oplus S_{15}$, | $S_{14} = M_4 \oplus A_V^+$, | |
| $K_{sa} = h(n_1, n_{4a}^*, P_{K_s})$, | $S_{15} = M_4 \oplus TID_V^+$, $S_{16} = n_4 \oplus B_V$, | |
| | $S_{17} = h(S_{14}, S_{15}, S_{16}, S_{17}, t_6)$ | |
| $P_{Ka_s}^+ = h(n_1, n_{4a}^*, K_{sa})$ | Replaces $TID_R$ with $TID_R^+$ | |
| Replaces $\{TID_V, A_V, P_{K_s}\}$ | $\{S_{14}, S_{15}, S_{16}, S_{17}, t_6\}$ $\longleftarrow$ | |
| with $\{TID_{Va}^+, A_{Va}^+, P_{Ka_s}^+\}$ | | |

**Figure 1.** Authentication and key agreement phase of Xu et al.'s scheme.

(1) The vehicle first generates a nonce $n_1$ and timestamp $t_1$, and computes $B_V = h(ID_V, P_{K_s})$, $S_1 = n_1 \oplus B_V$, and $S_2 = h(ID_V, TID_V, A_V, S_1, t_1, n_1)$. Then, the vehicle sends messages for authentication $\{TID_V, A_V, S_1, S_2, t_1\}$ to the RSU.

(2) After receiving the messages, the RSU checks the freshness of timestamp $t_1$. If it is valid, the RSU generates nonce $n_2$ and a timestamp $t_2$, and then the RSU computes $S_3 = n_2 \oplus B_R$, $S_4 = h(TID_V, TID_R, ID_R, S_3, t_2, n_2)$, and sends messages $\{(TID_V, TID_R, S_3, S_4, t_2)\}$ to the TA.

(3) The TA checks the freshness of timestamp $t_2$. If it is verified, the TA checks if $\{TID_V, TID_R\}$ are in the database. Then, the TA retrieves the corresponding data $\{TID_R, ID_R, B_R\}$ and computes $n_2^* = S_3 \oplus B_R$, $S_4^* = h(TID_V, TID_R, ID_R, S_3, t_2, n_2^*)$.

(4) The TA checks whether $S_4^* \stackrel{?}{=} S_4$, and if it is correct, the TA generates a nonce $n_3$ and a timestamp $t_3$. After that, the TA computes $M_1 = h(n_2^*, n_3, K_{TA})$, $S_5 = n_3 \oplus B_R$, $S_6 = M_1 \oplus P_{K_s}$, $S_7 = h(ID_R, S_5, S_6, X_V, n_3, t_3)$, and sends the messages $\{S_5, S_6, S_7, X_V, t_3\}$ to the RSU.

(5) On receiving the messages for the TA, the RSU checks the freshness of timestamp $t_3$, and then computes $n_3^* = S_5 \oplus B_R$ and $S_7^* = h(ID_R, S_5, S_6, X_V, n_3^*, t_3)$. The RSU also checks whether $S_7^* \stackrel{?}{=} S_7$. If it is correct, the RSU computes $M_1 = h(n_2, n_3^*, K_{TA})$, $P_{K_s} = M_1 \oplus S_6$, $r^* = A_V \oplus K_{TA}$, $ID_V^* = X_V \oplus h(r^*, K_{TA})$, $B_V = h(ID_V^*, P_{K_s})$, $n_1^* = S_1 \oplus P_{K_s}$, and $S_2^* = h(ID_V^*, TID_V, A_V, S_1, t_1, n_1^*)$. After that, the RSU generates a nonce $n_4$, $r^+$ and timestamp $t_4$, and checks if $S_2^* \stackrel{?}{=} S_2$. If it is valid, the RSU computes $K_s = h(n_1^*, n_4, P_{K_s})$, $P_{K_s}^+ = h(n_1^*, n_4, K_s)$, $X_V^+ = ID_V^* \oplus h(r^+, K_{TA})$, $S_8 = n_2 \oplus M_1 \oplus P_{K_s}^+$, $S_9 = n_3^* \oplus M_1, \oplus X_V^+$, $S_{10} = h(S_8, S_9, K_{TA}, n_2, n_3^*, t_4)$, and sends these messages $\{S_8, S_9, S_{10}, t_4\}$ to the TA.

(6) TA checks the freshness of timestamp $t_4$. If it is correct, the TA computes $S_{10}^* = h(S_8, S_9, K_{TA}, n_2^*, n_3, t_4)$ and checks if $S_{10}^* \stackrel{?}{=} S_{10}$. The TA computes $P_{K_s}^+ = n_2^* \oplus M_1 \oplus S_8$, $X_V^+ = n_3 \oplus M_1 \oplus S_9$ and selects a new identity $TID_V^+$, a new temporary identity $TID_R^+$, and a timestamp $t_5$. Then, the TA calculates $M_2 = h(n_2^*, n_3, P_{K_s})$, $M_3 = h(ID_R, n_2^*, n_3)$, $S_{11} = M_2 \oplus TID_V^+$, $S_{12} = M_3 \oplus TID_R^+$, $S_{13} = h(S_{11}, S_{12}, K_{TA}, M_2, M_3, t_5)$ and replaces $\{TID_V, X_V, P_{K_s}\}$ and $\{TID_R, ID_R, B_R\}$ with $\{TID_V, X_V, P_{K_s}, TID_V^+, X_V^+, P_{K_s}^+\}$ and $\{TID_R, ID_R, B_R, TID_R^+\}$ in database. The TA also sends the messages $\{S_{11}, S_{12}, S_{13}, t_5\}$ to the RSU.

(7) After receiving the messages, the RSU checks the freshness of timestamp $t_5$. If it is satisfied, the RSU computes $M_2^* = h(n_2, n_3^*, P_{K_s})$, $M_3^* = h(ID_R, n_2, n_3^*)$, $S_{13}^* = h(S_{11}, S_{12}, K_{TA}, M_2^*, M_3^*, t_5)$, and then checks whether $S_{13}^* \stackrel{?}{=} S_{13}$. If it is valid, the RSU computes a timestamp $t_6$, $TID_V^+ = M_2^* \oplus S_{11}$, $TID_R^+ = M_3^* \oplus S_{12}$, $A_V^+ = r^+ \oplus K_{TA}$, $M_4 = h(n_1^*, n_4, ID_V^*)$, $S_{14} = M_4 \oplus A_V^+$, $S_{15} = M_4 \oplus TID_V^+$, $S_{16} = n_4 \oplus B_V$, $S_{17} = h(S_{14}, S_{15}, S_{16}, S_{17}, t_6)$ and replaces $TID_R$ with $TID_R^+$ in the memory. The RSU sends the messages $\{S_{14}, S_{15}, S_{16}, S_{17}, t_6\}$ to the vehicle.

(8) The vehicle checks the freshness of timestamp $t_6$. If it is valid, the vehicle computes $n_4^* = S_{16} \oplus B_V$, $S_{17}^* = h(S_{14}, S_{15}, S_{16}, ID_V, n_4^*, t_6)$, and then checks if $S_{17}^* \stackrel{?}{=} S_{17}$. If it is true, the vehicle computes $M_4 = h(n_1, n_4^*, ID_V)$, $A_V^+ = M_4 \oplus S_{14}$, $TID_V^+ = M_4 \oplus S_{15}$, $K_s = h(n_1, n_4^*, P_{K_s})$, and $P_{K_s}^+ = h(n_1, n_4^*, K_s)$. Finally, the vehicle replaces the parameters $\{TID_V, A_V, P_{K_s}\}$ with the updated parameter $\{TID_V^+, A_V^+, P_{K_s}^+\}$ in its memory.

## 3. Security Flaws of Xu et al.'s Scheme

In this section, we prove that Xu et al.'s scheme is insecure against physical capture and impersonation attacks. Moreover, we also demonstrate that Xu et al.'s scheme is unable to guarantee the essential security requirements such as session key security and secure mutual authentication.

### *3.1. Formal Analysis Using ROR Model*

We show that Xu et al.'s scheme cannot ensure the security of the session key using formal security analysis as the "ROR model [14]", which is generally the accepted formal proof method [15,16]. To evaluate its security, we first describe the fundamentals of this model, and then analyze the security of Xu et al.'s scheme through this proof.

- Participants: The instances $inst_1$, $inst_2$, and $inst_3$ of the protocol for the vehicle (VH), trust authority (TA), and roadside unit (RSU) are represented by $\Pi_{VH}^{inst_1}$, $\Pi_{TA}^{inst_1}$, and $\Pi_{RSU}^{inst_2}$, respectively.
- Accepted state: Upon completion of the message exchange, the oracle $\Pi^{inst}$ enters an accepted state, and its current session identifier *sid* is defined by the ordered concatenation of all exchanged messages.
- Partnering: $\Pi_{VH}^{inst_1}$, $\Pi_{TA}^{inst_2}$, and $\Pi_{RSU}^{inst_3}$ are defined as partners when they share the same session identifier (*sid*), reach the accepted state, and successfully complete a mutual AKE.
- Freshness: To carry out the ROR proof, the instances ($\Pi_{VH}^{inst_1}$, $\Pi_{TA}^{inst_2}$, and $\Pi_{RSU}^{inst_3}$) are considered fresh if the current session key shared between the VH, TA, and RSU remains uncompromised by attacker $A$.
- Attacker: According to Xu et al.'s threat model [11], an $A$ can completely control the public networks and utilize the ROR queries described in Table 2 to break its security.
- Semantic Security: $A$ tries to guess the session key of instance from a random nonce. $A$ first guesses a bit $c$ using ROR queries. If $A$ can correctly guess a bit $c$, $A$ wins the game and breaks the semantic security of the scheme. The event of adversary $A$ winning the game is denoted as "$Win$", with the breaking advantage of the session key of Xu et al.'s scheme $P$ defined as $Adv_P = |2Pr[Win] - 1|$.
- Random Oracle: All entities can have access to a Random Oracle, which is implemented as a collision-resistant hash function $H$.

**Table 2.** Queries and its descriptions.

| Queries | Descriptions |
| --- | --- |
| $Execute(\Pi_{UE}^{inst_1}, \Pi_{SN}^{inst_2})$ | This query simulates an eavesdropping attack where adversary $A$ can intercept and monitor messages exchanged over the public network. |
| $CorruptUE(\Pi_{VH}^{inst_1})$ | This query represents a physical capture attack, allowing adversary $A$ to extract data stored in the vehicle VH. |
| $Send(\Pi^{inst}, Msg)$ | This query simulates an active attack where adversary $A$ can transmit a message to the oracle $P^inst$ and receive its response. |
| $Test(\Pi^{inst})$ | This query models a challenge to the security of a fresh session key $SK$, where based on a random coin flip $c$, the adversary $A$ receives either the real session key ($c = 1$) or a random value ($c = 0$), or NULL ($\perp$) if the key is not fresh. |

We demonstrate that Xu et al.'s scheme is unable to guarantee the security of a session key by the Theorem 1.

**Theorem 1.** *Under the assumption that the TA remains uncompromised, we define $q_h$, $q_s$, $q_e$, $N$, and $l_s$ as the number of Random Oracle queries, Send queries, Execute queries, the range space of random numbers, and the hash function output length, respectively. Let adversary $A$, operating within polynomial time $t$ against Xu et al.'s scheme $P$, have an advantage $Adv_P^A$ in compromising the security of session key. Then,*

$$Adv_{\mathcal{P}}^{A} \geq \frac{q_h^2}{2^{l_s}} + \frac{(q_s + q_e)^2}{N} \tag{1}$$

The Real-or-Random (ROR) model-based formal proofs are performed around a series of games $G_i (i = 0, 1, 2)$, where $Succ_i$ denotes the event in which adversary $A$ successfully wins game $G_i$.

- Game $G_0$: Game $G_0$ simulates an active attack by adversary $A$ against the protocol, with a randomly selected $c$ at the game's outset and a specific winning advantage.

$$Adv_{\mathcal{P}}^{A} = |2.Pr[Succ_0] - 1| \tag{2}$$

- Game $G_1$: Game $G_1$ models an eavesdropping attack where $A$ controls exchanged messages via *Execute* queries, followed by a *Test* query to distinguish between the actual $SK$ and a arbitrary value. In Xu et al.'s scheme, the VH, TA, and RSU make an $SK$ computed by $K_s = h(n_1, n_4, P_{Ks})$. Although $A$ obtains all the messages transmitted in public channel, $A$ does not compute the $SK$ and increase the probability of $A$ winning games. Then, we could obtain

$$Pr[Succ_1] = Pr[Succ_0] \tag{3}$$

- Game $G_2$: Game $G_2$ models an impersonation attack, where $A$ attempts to impersonate the legitimate $VH$ using $Send(\Pi^{inst}, Msg)$, $CorruptUE(\Pi_{VH}^{inst_1})$, and some $Hash$ queries. Under the threat model of Xu et al., $A$ first executes the $CorruptUE(\Pi_{VH}^{inst_1})$ query, and then obtains the value $\{ID_V, TID_V, A_V, P_{K_s}\}$ in the vehicle's memory. The collision probability for hash and random number are given by $\frac{q_h^2}{2^{l_s+1}}$ and $\frac{(q_s+q_e)^2}{2N}$, respectively. $A$ can break the $SK$ by utilizing the obtained real identity $ID_V$, $P_{K_s}$, and $A_V$ because $A$ can execute the AKE procedure without needing to resolve the hash collision problem. Therefore, Game $G_1$ and $G_2$ are distinguishable. Then, we could obtain the following:

$$|Pr[Succ_1] - Pr[Succ_2| \geq \frac{q_h^2}{2^{l_s+1}} + \frac{(q_s + q_e)^2}{2N} \tag{4}$$

Upon completion of all games $(G_0, G_1, G_2)$, $A$ attempts to correctly determine the $c$ using *Test* query. Therefore,

$$Adv_{\mathcal{P},G_2}^{A} = \frac{1}{2} \tag{5}$$

The result can be derived using Equations (2), (3), and (5).

$$\begin{aligned} \frac{1}{2}.Adv_{\mathcal{P}}^{A} &= |Pr[Succ_0] - \frac{1}{2}| \\ &= |Pr[Succ_1] - \frac{1}{2}| \\ &= |Pr[Succ_1] - Pr[Succ_2]| \end{aligned} \tag{6}$$

Then, the final result can be derived using Equations (4)–(6):

$$Adv_{\mathcal{P}}^{A} \geq \frac{q_h^2}{2^{l_s}} + \frac{(q_s + q_e)^2}{N} \tag{7}$$

Finally, we can break the $SK$ without solving the problem of hash collision. Therefore, we demonstrate that Xu et al.'s scheme does not ensure the $SK$ security using the ROR proof.

### *3.2. Informal Security Analysis*

We point out that Xu et al.'s scheme is unable to resist physical capture and impersonation attacks, and achieve secure mutual AKE using this analysis.

#### 3.2.1. Physical Capture Attack

During the registration procedure, the TA stores the values $\{ID_V, TID_V, A_V, P_{K_s}\}$ in the memory of VH without adopting any cryptographic methods. Moreover, under the threat model of Xu et al.'s scheme (Section 1.2), an adversary $A$ can easily extract these values by power analysis attacks [18,19]. Based on this attack, $A$ could easily perform impersonation attacks and compromise the privacy of users.

#### 3.2.2. Vehicle Impersonation Attack

In Xu et al.'s scheme [11], they analyze its security using the proposed threat model. Unfortunately, if an adversary $A$ compromises the vehicle and obtains the data of the vehicle's memory using the physical capture attacks, $A$ can easily generate the request messages $\{TID_V, A_V, S_1, S_2\}$ and calculate the correct session key $K_s = h(n_1, n_4^*, K_s)$ because the private parameters of vehicle's memory is stored without applying any cryptographic techniques. Therefore, Xu et al.'s scheme does not resist impersonation attacks, and its description of the processes included in this attack is presented in Figure 2.

#### 3.2.3. Secure Mutual Authentication

According to the above-mentioned attacks, $A$ can easily impersonate a legitimate vehicle to access the IoV networks presented by Xu et al.'s scheme and successfully authenticate among participated nodes. Moreover, $A$ can compute the session key between vehicle/RSU and TA. Therefore, Xu et al.'s scheme is unable to ensure a mutual AKE.

#### 3.2.4. Insecure Authentication Mechanism

In Xu et al.'s scheme, the vehicle sends the access request $\{TID_V, A_V, S_{1_a}, S_{2_a}, t_1\}$ and only receives final response messages $\{S_{14}, S_{15}, S_{16}, S_{17}, t_6\}$ without verification. However, it becomes highly vulnerable to various attacks because the vehicle is not physically protected. Therefore, it is essential to include steps in the three-party authentication process where messages exchanged with the vehicle during its processes can be verified.

#### 3.2.5. Correctness of Threat Model

Xu et al. indicate security assumptions to evaluate the protocol's security, and then asserted that the proposed scheme can resist against physical capture and impersonation attacks. Unfortunately, we proved that Xu et al.'s scheme is insecure against these attacks under Xu et al.'s threat model and they were unable to contemplate various security attacks. Therefore, we propose the solutions to enhance the security level in Section 4.

### *3.3. Comparative Analysis of Security Properties*

We perform a comparative analysis of the security properties of Xu et al.'s scheme and other related works [4–6,11,13]. Table 3 presents the security features of both Xu et al.'s and prior schemes. The earlier schemes failed to prevent multiple types of attacks. To prevent these attacks, we suggest the solutions in Section 4.

**Table 3.** Security property of Xu et al.'s scheme with other related schemes.

| Security Property | Ying and Nayak [5] | Chen et al. [6] | He et al. [4] | Wang et al. [13] | Xu et al. [11] |
|---|---|---|---|---|---|
| $SP_1$ | × | × | × | ○ | × |
| $SP_2$ | × | ○ | ○ | ○ | × |
| $SP_3$ | ○ | ○ | ○ | ○ | ○ |
| $SP_4$ | × | × | × | ○ | × |
| $SP_5$ | ○ | ○ | ○ | ○ | ○ |
| $SP_6$ | × | ○ | ○ | ○ | ○ |
| $SP_7$ | × | × | × | N/A | ○ |

○: the security properties are preserved; ×: the security properties are not preserved; N/A: not applicable; $SP_1$: vehicle capture attack; $SP_2$: replay attack; $SP_3$: vehicle anonymity; $SP_4$: vehicle impersonation; $SP_5$: anonymity of RSU; $SP_6$: session fixation attack; $SP_7$: RSU-aided authentication function.

| **Adversary** $(A/VH_i)$ | **Roadside unit** (RSU) | **Trusted authority** (TA) |
|---|---|---|
| Generates $n_1$, timestamp $t_1$<br>Computes $B_{V_a} = h(ID_V, P_{K_s})$,<br>$S_{1_a} = n_1 \oplus B_{V_a}$<br>$S_{2_a} = h(ID_V, TID_V, A_V, S_{1_a}, t_1, n_1)$<br>$\underrightarrow{\{TID_V, A_V, S_{1_a}, S_{2_a}, t_1\}}$ | | |
| | Checks validity of $t_1$<br>Selects $n_2$, timestamp $t_2$<br>Computes $S_3 = n_2 \oplus B_R$,<br>$S_4 = h(TID_V, TID_R, ID_R, S_3, t_2, n_2)$<br>$\underrightarrow{\{(TID_V, TID_R, S_3, S_4, t_2)\}}$ | |
| | | Checks validity of $t_2$<br>Checks if $TID_V, TID_R$ are in database.<br>Retrieves $\{TID_V, X_V, P_{K_s}\}$,<br>$\{TID_R, TID_R, B_R\}$, respectively.<br>Checks $S_4^* \stackrel{?}{=} S_4$<br>Generates nonce $n_3$, timestamp $t_3$<br>Computes $M_1 = h(n_2^*, n_3, K_{TA})$,<br>$S_5 = n_3 \oplus B_R$, $S_6 = M_1 \oplus P_{K_s}$,<br>$S_7 = h(ID_R, S_5, S_6, X_V, n_3, t_3)$<br>$\underleftarrow{\{S_5, S_6, S_7, X_V, t_3\}}$ |
| | Checks validity of $t_3$<br>Computes $n_3^* = S_5 \oplus B_R$,<br>$S_7^* = h(ID_R, S_5, S_6, X_V, n_3^*, t_3)$,<br>Checks $S_7^* \stackrel{?}{=} S_7$<br>Computes $M_1 = h(n_2, n_3^*, K_{TA})$,<br>$P_{K_s} = M_1, \oplus S_6, r^* = A_V \oplus K_{TA}$,<br>$ID_V^* = X_V \oplus h(r^*, K_{TA})$,<br>$B_V = h(ID_V^*, P_{K_s}), n_1^* = S_1 \oplus P_{K_s}$,<br>$S_2^* = h(ID_V^*, TID_V, A_V, S_1, t_1, n_1^*)$<br>Checks $S_2^* \stackrel{?}{=} S_2$<br>Generates $n_4$, $r^+$ and timestamp $t_4$,<br>Computes $K_s = h(n_1^*, n_4, P_{K_s})$,<br>$P_{K_s}^+ = h(n_1^*, n_4, K_s)$,<br>$X_V^+ = ID_V^* \oplus h(r^+, K_{TA})$,<br>$S_8 = n_2 \oplus M_1 \oplus P_{K_s}^+$,<br>$S_9 = n_3^* \oplus M_1, \oplus X_V^+$<br>$S_{10} = h(S_8, S_9, K_{TA}, n_2, n_3^*, t_4)$<br>$\underrightarrow{\{S_8, S_9, S_{10}, t_4\}}$ | Checks validity of $t_4$<br>Computes $S_{10}^* = h(S_8, S_9, K_{TA}, n_2^*, n_3, t_4)$<br>Checks $S_{10}^* \stackrel{?}{=} S_{10}$<br>Computes $P_{K_s}^+ = n_2^* \oplus M_1 \oplus S_8$,<br>$X_V^+ = n_3 \oplus M_1 \oplus S_9$<br>Selects $TID_V^+, TID_R^+$, timestamp $t_5$<br>Calculates $M_2 = h(n_2^*, n_3, P_{K_s})$,<br>$M_3 = h(ID_R, n_2^*, n_3)$, $S_{11} = M_2 \oplus TID_V^+$,<br>$S_{12} = M_3 \oplus TID_R^+$,<br>$S_{13} = h(S_{11}, S_{12}, K_{TA}, M_2, M_3, t_5)$,<br>replaces $\{TID_V, X_V, P_{K_s}\}$ and $\{TID_R, ID_R, B_R\}$<br>with $\{TID_V, X_V, P_{K_s}, TID_V^+, X_V^+, P_{K_s}^+\}$<br>and $\{TID_R, ID_R, B_R, TID_R^+\}$, respectively.<br>$\underleftarrow{\{S_{11}, S_{12}, S_{13}, t_5\}}$ |
| Checks validity of $t_6$<br>Computes $n_{4a}^* = S_{16} \oplus B_V$,<br>$S_{17}^* = h(S_{14}, S_{15}, S_{16}, ID_V, n_4^*, t_6)$<br>Checks $S_{17}^* \stackrel{?}{=} S_{17}$<br>Computes $M_4 = h(n_1, n_{4a}^*, ID_V)$,<br>$A_{Va}^+ = M_{4_a} \oplus S_{14}$,<br>$TID_{Va}^+ = M_{4_a} \oplus S_{15}$,<br>$K_{sa} = h(n_1, n_{4a}^*, P_{K_s})$,<br>$P_{Ka_s}^+ = h(n_1, n_{4a}^*, K_{sa})$<br>Replaces $\{TID_V, A_V, P_{K_s}\}$<br>with $\{TID_{Va}^+, A_{Va}^+, P_{Ka_s}^+\}$ | Checks validity of $t_5$<br>Computes $M_2^* = h(n_2, n_3^*, P_{K_s})$,<br>$M_3^* = h(ID_R, n_2, n_3^*)$,<br>$S_{13}^* = h(S_{11}, S_{12}, K_{TA}, M_2^*, M_3^*, t_5)$<br>Checks $S_{13}^* \stackrel{?}{=} S_{13}$<br>Generates timestamp $t_6$<br>Computes $TID_V^+ = M_2^* \oplus S_{11}$,<br>$TID_R^+ = M_3^* \oplus S_{12}$, $A_V^+ = r^+ \oplus K_{TA}$,<br>$M_{4_a} = h(n_1^*, n_4, ID_V^*)$,<br>$S_{14} = M_4 \oplus A_V^+$,<br>$S_{15} = M_4 \oplus TID_V^+$, $S_{16} = n_4 \oplus B_V$,<br>$S_{17} = h(S_{14}, S_{15}, S_{16}, S_{17}, t_6)$<br>Replaces $TID_R$ with $TID_R^+$<br>$\underleftarrow{\{S_{14}, S_{15}, S_{16}, S_{17}, t_6\}}$ | |

**Figure 2.** Vehicle impersonation attack in Xu et al.'s scheme.

### *3.4. Simulation Analysis Using AVISPA Tool*

The security of the proposed cryptographic protocol is formally validated using AVISPA, a widely accepted tool for analyzing security protocols [20,21]. AVISPA specifically verifies protocols against replay and man-in-the-middle attacks. The tool utilizes the High-Level Protocol Specification Language (HLPSL) [22] to define the security features of protocols. The AVISPA employs four back-end models [23]: "constraint logic-based attack searcher (CL-AtSE)", "on the fly model checker (OFMC)", "SAT-based model checker (SATMC)", and "tree automata based on protocol analyzer (TA4SP)". The process involves translating the HLPSL code into an Intermediate Format (IF) using the "HLPSL2IF" translator, and then it is analyzed by the four back-ends to assess security properties. Figure 3 illustrates this process, and a detailed description of HLPSL can be found in [21,22].

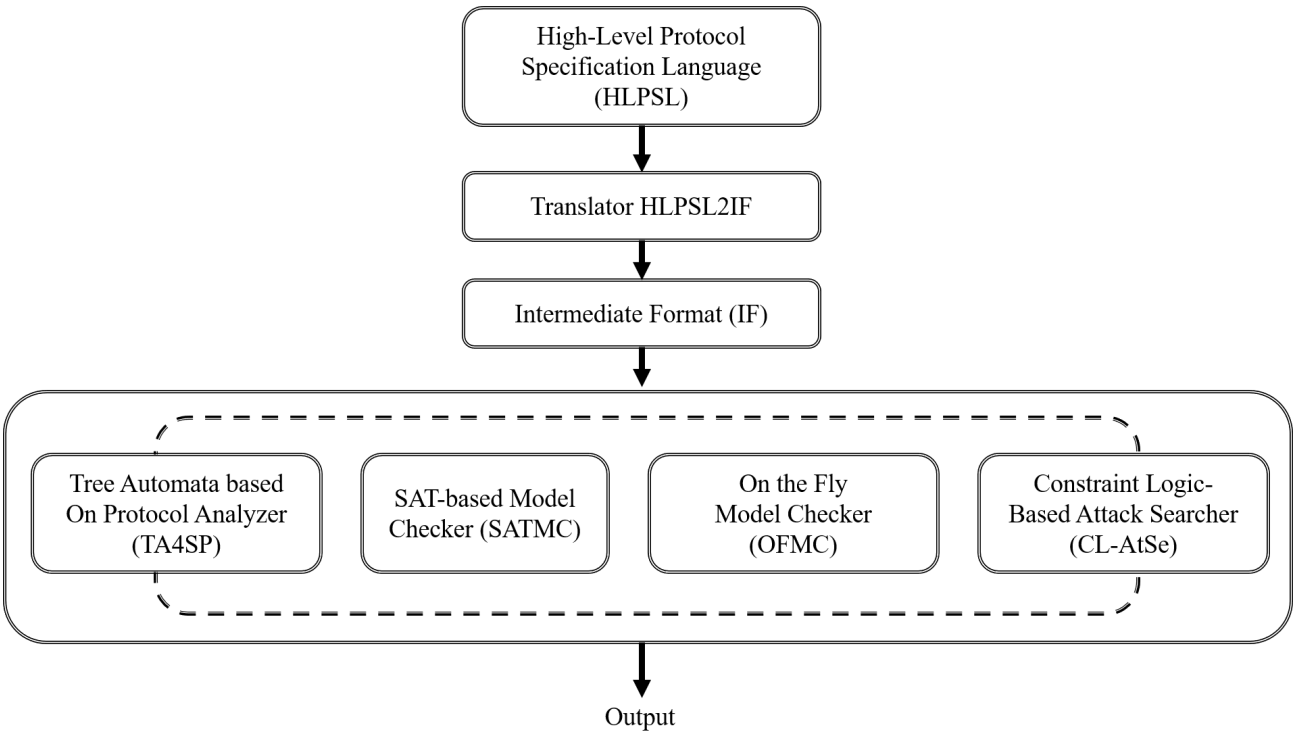



**Figure 3.** The process of AVISPA simulation.

#### 3.4.1. Simulation Environments

The AVISPA simulation is operated on a system running on Ubuntu 10.10, equipped with 2 GB of RAM, and powered by an Intel Core I9-11900K processor with a clock speed of 3.50 GHz and 64 GB of RAM.

#### 3.4.2. HLPLS Specifications

To evaluate the security of protocols through this simulation, all phases of Xu et al.'s scheme are presented using HLPLS code. We have tested it by considering the authentication phases among entities. In Xu et al.'s scheme, there are three basic roles (TA, VH, RSU), and their descriptions are illustrated in Figures 4–6. The environment and session are shown in Figure 7.

```
role trustparty(VH,RSU,TA: agent, SKvta,SKrta:symmetric_key, H:hash_func, SND,RCV: channel(dy)) played_by TA
def=
        local State:nat,
        IDv,TIDv,Av,Kta,PKs,R,TIDr,IDr,Br,Xv,Bv,N1,N2,N3,N4: text,
        T1,T2,T3,T4,T5,T6,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S15,S17: text,
        RP,Ks,Xvp,PKsp,TIDvp,TIDrp,M1: text
        const sp1,sp2,v_t_idv,v_r_n1,r_v_n4: protocol_id
init State := 0
transition
        1. State=0 /\RCV(start) =|>
        State':=1 /\R':=new()
        /\TIDv':=new()
        /\PKs':=new()
        /\IDv':=new()
        /\Av':=xor(R',Kta)
        /\SND({IDv'.TIDv'.Av'.PKs'}_SKvta)
        /\secret({IDv',Av',PKs'},sp1,{TA,VH})
        2. State=1 /\ RCV(TIDv.xor(R',Kta).S1'.S2'.T1') =|>
        State':=2 /\TIDr':=new()
        /\IDr':=new()
        /\Br':=H(IDr'.Kta)
        /\Xv':=xor(IDv,H(R'.Kta))
        /\SND({IDr'.TIDr'.Br'.Kta}_SKrta)
        /\secret({IDr',Br',Kta},sp2,{TA,RSU})

        3. State=2  /\RCV(TIDv.TIDr.xor(N2',H(IDr'.Kta)).H(TIDv.TIDr'.IDr'.S3'.T2'.N2').T2') =|>
        State':=3 /\N3':=new()
        /\T3':=new()
        /\M1':=H(N2'.N3'.Kta)
        /\S5':=xor(N3',Br)
        /\S6':=xor(M1',PKs)
        /\S7':=H(IDr.S5'.S6'.Xv.N3'.T3')
        /\SND(S5'.S6'.S7'.Xv.T3')
        4. State=3
/\RCV(xor(N2,xor(H(N2'.N3'.Kta),H(N1'.N4'.Ks'))).xor(N3',xor(H(N2'.N3'.Kta),xor(IDv',H(RP'.Kta)))).H(xor(N2,xor(H(N2'
.N3'.Kta),H(N1'.N4'.Ks'))).xor(N3',xor(H(N2'.N3'.Kta),xor(IDv',H(RP'.Kta)))).Kta.N2.N3'.T4').T4') =|>
        State':=4 /\TIDvp':=new()
        /\TIDrp':=new()
        /\T5':=new()
        /\S11':=xor(H(N2.N3.PKs),TIDvp')
        /\S12':=xor(H(IDr.N2.N3),TIDrp')
        /\S13':=H(S11'.S12'.Kta.H(N2.N3.PKs).H(IDr.N2.N3),T5')
        /\SND(S11'.S12'.S13'.T5')

end role
```

**Figure 4.** Role: TA's HLPSL description.

```
role vehicle(VH,RSU,TA: agent, SKvta:symmetric_key, H:hash_func, SND,RCV: channel(dy)) played_by VH
def=
        local State:nat,
        IDv,TIDv,Av,Kta,PKs,R,TIDr,IDr,Br,Xv,Bv,N1,N2,N3,N4: text,
        T1,T2,T3,T4,T5,T6,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S15,S17: text,
        RP,Ks,Xvp,PKsp,TIDvp,TIDrp,M1: text
        const sp1,sp2,v_r_n1,r_v_n4: protocol_id

init State := 0
transition
        1. State=0 /\RCV({IDv'.TIDv'.xor(R',Kta).PKs'}_SKvta) =|>
        State':=1 /\N1':=new()
        /\T1':=new()
        /\Bv':=H(IDv'.PKs')
        /\S1':=xor(N1',Bv')
        /\S2':=H(IDv'.TIDv'.xor(R'.Kta).S1'.T1'.N1')
        /\SND(TIDv'.xor(R',Kta).S1'.S2'.T1')
        /\request(VH,RSU,v_r_n1,N1')
        2. State=1
/\RCV(xor(H(N1.N4'.IDv),xor(RP,Kta)).xor(xor(H(N1.N4'.IDv),TIDvp')).xor(N4',H(IDv.PKs)).H(xor(H(N1.N4'.IDv),xor(R
P,Kta)).xor(xor(H(N1.N4'.IDv),TIDvp')).xor(N4',H(IDv.PKs)).IDv.N4'.T6').T6') =|>
        State':=2 /\witness(RSU,VH,r_v_n4,N4')

end role
```

**Figure 5.** Role: VH's HLPSL description.

```
role rodasu(VH,RSU,TA: agent, SKrta:symmetric_key, H:hash_func, SND,RCV: channel(dy)) played_by RSU
def=
        local State:nat,
        IDv,TIDv,Av,Kta,PKs,R,TIDr,IDr,Br,Xv,Bv,N1,N2,N3,N4: text,
        T1,T2,T3,T4,T5,T6,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S15,S17: text,
        RP,Ks,Xvp,PKsp,TIDvp,TIDrp,M1: text
        const sp1,sp2,v_t_idv,v_r_n1,r_v_n4: protocol_id

init State := 0
transition
        1. State=0 /\RCV(TIDv'.xor(R',Kta).S1'.S2'.T1') =|>
        State':=1 /\secret({IDv,xor(R',Kta),PKs},sp1,{TA,VH})

        2. State=1 /\ RCV({IDr'.TIDr'.H(IDr'.Kta).Kta}_SKrta) =|>
        State':=2 /\secret({IDr',H(IDr'.Kta),Kta},sp2,{TA,RSU})
        /\N2':=new()
        /\T2':=new()
        /\S3':=xor(N2',H(IDr'.Kta))
        /\S4':=H(TIDv.TIDr'.IDr'.S3'.T2'.N2')
        /\SND(TIDv.TIDr'.S3'.S4'.T2')
        3. State=2 /\RCV(xor(N3',Br).xor(M1',PKs).S7'.xor(IDv',H(R'.Kta)).T3') =|>
        State':=3 /\N4':=new()
        /\RP':=new()
        /\T4':=new()
        /\Ks':=H(N1.N4'.PKs)
        /\PKsp':=H(N1.N4'.Ks')
        /\Xvp':=xor(IDv',H(RP'.Kta))
        /\S8':=xor(N2,xor(H(N2.N3'.Kta),PKsp'))
        /\S9':=xor(N3',xor(H(N2.N3'.Kta),Xvp'))
        /\S10':=H(S8'.S9'.Kta.N2.N3'.T4')
        /\SND(S8'.S9'.S10'.T4')
        /\request(VH,RSU,v_r_n1,N1)
        4. State=3
/\RCV(xor(H(N2.N3.PKs),TIDvp').xor(H(IDr.N2.N3),TIDrp').H(xor(H(N2.N3.PKs),TIDvp').xor(H(IDr.N2.N3),TIDrp').Kta.
H(N2.N3.PKs).H(IDr.N2.N3),T5').T5') =|>
        State':=4 /\T6':=new()
        %/\M4':=H(N1.N4.IDv)
        %/\Avp':=xor(RP,Kta)
        %/\S14':=xor(H(N1.N4.IDv),xor(RP,Kta))
        /\S15':=xor(xor(H(N1.N4.IDv),TIDvp'))
        %/\S16':=xor(N4,H(IDv.PKs'))
        /\S17':=H(xor(H(N1.N4.IDv),xor(RP,Kta)).S15'.xor(N4,H(IDv.PKs)).IDv.N4.T6')
        /\SND(xor(H(N1.N4.IDv),xor(RP,Kta)).S15'.xor(N4,H(IDv.PKs)).S17'.T6')
        /\witness(RSU,VH,r_v_n4,N4)

end role
```

**Figure 6.** Role: RSU's HLPSL description.

```
%%%%%%%%%%  Session  %%%%%%%%%
role session(VH,RSU,TA:agent, SKvta,SKrta:symmetric_key, H:hash_func)
def=
        local
                SN1,RV1,SN2,RV2,SN3,RV3:channel(dy)
        composition
    trustparty(VH,RSU,TA,SKvta,SKrta,H,SN1,RV1) /\
                rodasu(VH,RSU,TA, SKrta, H, SN2,RV2) /\
    vehicle(VH,RSU,TA, SKvta, H, SN3,RV3)
end role

%%%%%%%%%%  Environment  %%%%%%%%%
role environment()
def=
        const
                skvta,skrta:symmetric_key, vh,rsu,ta:agent,
                h:hash_func,
                idv,tidv,av,pks:text,
    sp1,sp2,v_r_n1,r_v_n4:protocol_id
        intruder_knowledge = {vh,rsu,ta,skvta,h,idv,tidv,av,pks}
        composition
                session(vh,rsu,ta,skvta,skrta,h)

end role

goal
        secrecy_of sp1,sp2
    authentication_on v_r_n1
    authentication_on r_v_n4
end goal

environment()
```

**Figure 7.** Session and environment: HLPSL description.

#### 3.4.3. Simulation Results

To prove that Xu et al.'s scheme is vulnerable to replay and man-in-the-middle attacks, we simulated the OMFC and CL-AtSe utilizing the above-mentioned codes (Figures 4–7), as follows:

- OFMC: The search depth and explored nodes are 1 and 3, respectively. The search time is 0.14 s.
- CL-AtSe: The time of translation is 0.02 s and 2 states are analyzed. the two states.

Figure 8 displays the results from the OFMC and CL-AtSe, indicating "UNSAFE". Consequently, Xu et al.'s scheme cannot prevent replay and man-in-the-middle attacks.

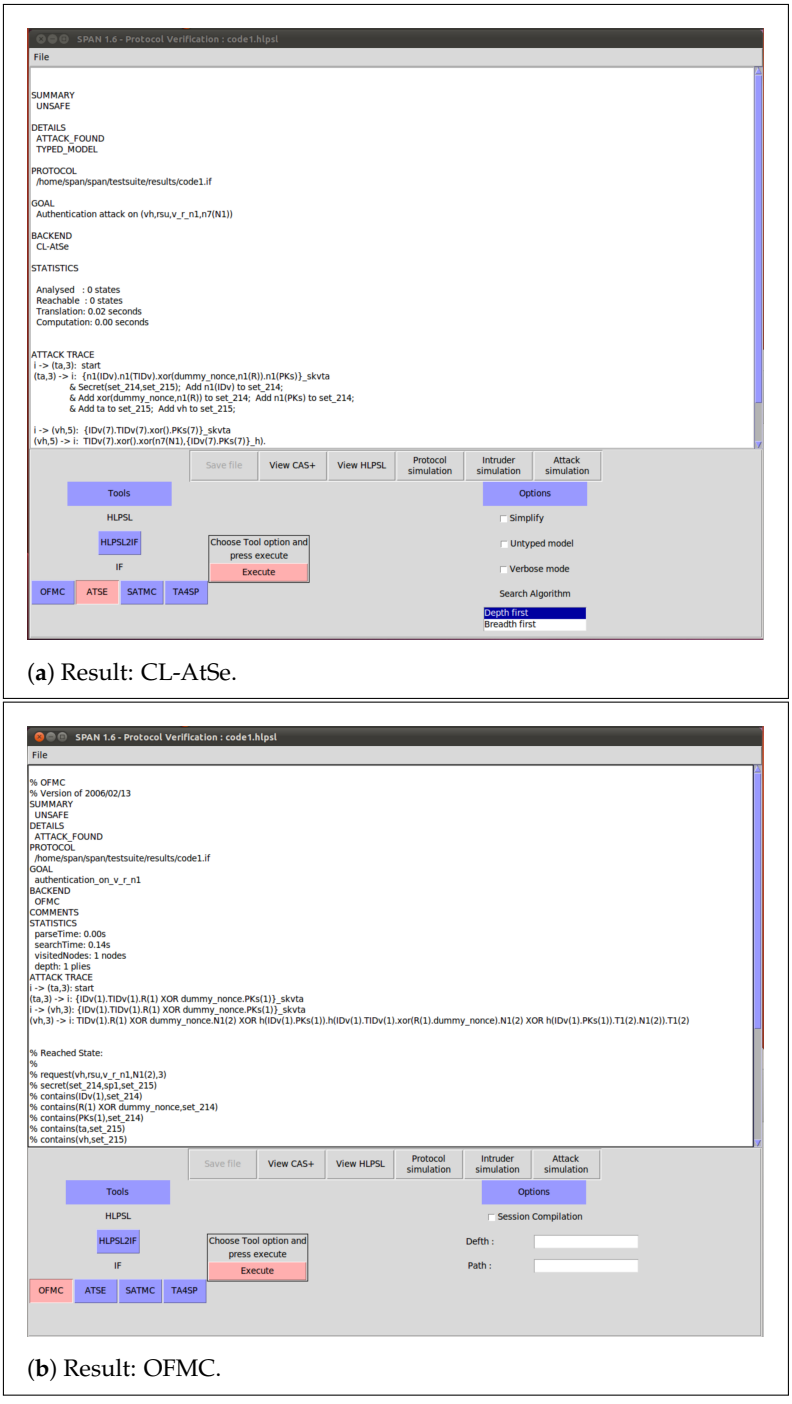



**Figure 8.** Simulation results.

## 4. Security Fixes

In Xu et al.'s scheme [11], the significant security vulnerabilities are that the secret information is stored in vehicles without proper cryptographic methods. Through these security weaknesses, an adversary can easily obtain the secret data stored in the OBU of vehicles, which are utilized to impersonate and make the session key among participated nodes. These significant security issues are presented in Section 3.

Over recent decades, numerous AKE schemes have been developed to protect user and data privacy. These schemes generally store secret information in the device memory for AKE between entities. Xu et al. also tried to resolve issues using the strong assumptions that the RSU and TA are trusted entities. Despite recognizing this vulnerability, Xu et al. failed to adequately resolve these concerns and did not consider all possible weaknesses in the design of their scheme. To alleviate the security issues identified in Xu et al.'s scheme, we suggest the following essential guidelines.

Fix 1. In the registration phase of Xu et al.'s scheme, the TA should not store the secret data $\{ID_V, TID_V, A_V, P_{K_s}\}$ as plaintext to prevent physical capture attacks. According to their threat model, an adversary could easily capture the OBU of the vehicle and obtain the stored data. The vehicle should securely store these parameters by encrypting them with XOR operations and a hash function. For instance, rather than storing $ID_V$ directly in the OBU, it can be stored by applying an XOR operation with specific parameters ($ID_V \oplus h(Password)$) or by generating parameters based on the hashed value of $ID_V$ ($TID_V = h(ID_V, Password, randomnumber)$). This method prevents attackers from easily accessing $ID_V$ and other secret parameters because these parameters are not stored in the OBU as plaintext.

Fix 2. To prevent physical capture attacks, we can employ the PUF on vehicles. In Xu et al.'s scheme, the identity and secret parameters are vulnerable to extraction, which facilitates the impersonation attacks. However, the PUF base system offers a robust solution by generating secret parameters that are inherently resistant to extraction [12,24]. For example, it is possible to generate session keys by generating parameters using the unique random values generated by a PUF. In Xu et al.'s method, the session key is generated using the random nonces $\{n_1, n_4\}$ and the hash value of the previous session key $P_{K_s}$. Therefore, if the $h(PUF)$ value is used in generating the session key ($K_s = h(PUF), n_1, n_4, P_{K_s}$), an attacker cannot steal the session key without access to the PUF device. This solution not only resolves the current security issues but also mitigates other potential vulnerabilities using the unique physical characteristics of PUF.

Fix 3. Xu et al.'s scheme adopted a two-factor authentication mechanism utilizing vehicles and secret information. However, the two-factor authentication mechanism used in the proposed scheme does not function properly if secret information is leaked. To address this limitation and enhance overall security, we suggest utilizing a three-factor authentication mechanism, which would incorporate biometric verification using a fuzzy extractor [25] with vehicle and PUF-based secret parameters. A fuzzy extractor, which is used in biometric-based authentication methods, can be utilized to design a three-factor AKE scheme. In a three-factor AKE scheme, secret parameters can be stored by conducting an XOR or hashing them with the biometric value generated by the fuzzy extractor. Therefore, even if an attacker physically captures the device, they cannot access the actual secret parameters used in the AKE without the biometric data. Therefore, we prevent the physical capture attacks.

The suggested solutions aim to mitigate vehicle impersonation attacks and we do not claim that our solutions are complete against all potential security vulnerabilities. Nevertheless, these enhancements significantly improve system security and increase the complexity of potential attacks for an adversary.

Xu et al. made a commendable effort in constructing a secure and efficient key agreement scheme for IoV. However, they would have examined their scheme from multiple perspectives. The advancement of research in this field is characterized by diverse approaches from different researchers. This paper serves to highlight the ongoing need for developing secure and efficient AKE schemes in IoV environments.

## 5. Conclusions and Future Works

This paper refers to "a secure and computationally efficient AKE scheme for IoV". We pointed out that Xu et al.'s scheme fails to prevent vehicle impersonation attacks, thereby not achieving secure AKE and not meeting the security requirements of their threat model. Furthermore, we demonstrate through formal mathematical analysis using the ROR model that it does not guarantee session key security. We also perform the simulation proof using the AVISPA which is well known for formal verification tool. These security vulnerabilities make the scheme unsuitable and impractical for deployment in real-world environments. Therefore, we propose solutions to enhance security levels to develop a more secure and efficient AKE for IoV environments. Our future work involves designing a robust and lightweight AKE scheme that can be applied in various environments with resource-limited devices by utilizing the proposed solution.

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

**Funding:** This work was supported by the Gachon University research fund of 2023 (GCU-202400440001).

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

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

## References

1. Liang, W.; Xie, S.; Long, J.; Li, K.-C.; Zhang, D.; Li, K. A double PUF-based RFID identity authentication protocol in service-centric internet of things environments. *Inform. Sci.* **2019**, *503*, 129–147. [CrossRef]
2. Sharma, V. An energy-efficient transaction model for the blockchain-enabled internet of vehicles (IoV). *IEEE Commun. Lett.* **2019**, *23*, 246–249. [CrossRef]
3. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE Trans. Ind. Inf.* **2018**, *14*, 3690–7300. [CrossRef]
4. He, D.; Zeadally, S.; Kumar, N.; Wu, W. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. *IEEE Trans Inf. Forensics. Secur.* **2016**, *11*, 2052–2064. [CrossRef]
5. Ying, B.; Nayak, A.; Anonymous and lightweight authentication for secure vehicular networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 10626–10636. [CrossRef]
6. Chen, C.-M.; Xiang, B.; Liu, Y.; Wang, K.-H. A secure authentication protocol for internet of vehicles. *IEEE Access* **2019**, *7*, 12047–12057. [CrossRef]
7. Yu, S.; Lee, J.; Park, K.; Das A.K.; Park, Y. IoV-SMAP: Secure and efficient message authentication protocol for IoV in smart city environment. *IEEE Access* **2020**, *8*, 167875–167886. [CrossRef]
8. Cui, J.; Xu, W.; Zhong, H.; Zhang, J.; Xu, Y.; Liu, L. Privacy-preserving authentication using a double pseudonym for internet of vehicles. *Sensors* **2018**, *18*, 1453. [CrossRef] [PubMed]
9. Dua, A.; Kumar, N.; Das, A.K.; Susilo, W. Secure message communication protocol among vehicles in smart city. *IEEE Trans Veh. Technol.* **2018**, *67*, 4359–4373. [CrossRef]
10. Li, X.; Liu, T.; Obaidat, M.S.; Wu, F.; Vijayakumar, P.; Kumar, N. A lightweight privacy-preserving authentication protocol for VANETs. *IEEE Syst. J.* **2020**, *14*, 3547–3557. [CrossRef]
11. Xu, A.; Li, X.; Xu, J.; Liang, W.; Choo, K.-K.R. A secure and computationally efficient authentication and key agreement scheme for Internet of Vehicles. *Comput. Electr. Eng.* **2021**, *95*, 107409. [CrossRef]
12. Yu, S.; Park, K. PUF-based robust and anonymous authentication and key establishment scheme for V2G networks. *IEEE Internet Things J.* **2024**, *11*, 15450–15464. [CrossRef]
13. Wang, S.; Fan, Z.; Su, Y.; Zheng, B.; Liu, Z.; Dai, Y. A Lightweight, Efficient, and Physically Secure Key Agreement Authentication Protocol for Vehicular Networks. *Electronics* **2024**, *13*, 1418. [CrossRef]

14. Abdalla, M.; Fouque, P.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Les Diablerets, Switzerland, 23–26 January 2005; Lecture Notes in Computer Science (LNCS); pp. 65–84.
15. Park, K.; Park, Y. MIoT-CDPS: Complete decentralized privacy-preserving scheme for medical internet of things. *Internet Things* **2024**, *27*, 101250. [CrossRef]
16. Park, K.; Lee, J.; Das, A.K.; Park, Y. BPPS:Blockchain-Enabled Privacy-Preserving Scheme for Demand-Response Management in Smart Grid Environments. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 1719-1729. [CrossRef]
17. Dolev, D.; Yao, A. C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198. [CrossRef]
18. Eisenbarth, T.; Kasper, T.; Moradi, A.; Paar, C.; Salmasizadeh, M.; Shalmani, M. T. M. On the power of power analysis in the real world: A complete break of the KEELOQ code hopping scheme. In *Advances in Cryptology–CRYPTO*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 203–220.
19. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology–CRYPTO, Proceedings of the 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
20. Kim, M.; Park, K.; Park, Y. A reliable and privacy-preserving vehicular energy trading scheme using decentralized identifiers. *Mathematics* **2024**, *12*, 1450. [CrossRef]
21. Yu, S.; Lee, J.; Sutrala, A.K.; Das, A.K.; Park, Y. LAKA-UAV: Lightweight authentication and key agreement scheme for cloud-assisted Unmanned Aerial Vehicle using blockchain in flying ad-hoc networks, *Comput. Netw.* **2023**, *224*, 109612. [CrossRef]
22. Von Oheimb, D. The high-level protocol specification language HLPSL developed in the EU project avispa. In Proceedings of the APPSEM 2005 Workshop, Tallinn, Finland, 13–15 September 2005; pp. 1–2.
23. Vigano, L. Automated Security Protocol Analysis with the AVISPA Tool. *Electron. Notes Theor. Comput. Sci.* **2006**, *155*, 61. [CrossRef]
24. Yu, S.; Park, Y. Robust and efficient authentication and group—proof scheme using physical unclonable functions for wearable computing. *Sensors* **2023**, *23*, 5747. [CrossRef]
25. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A.; Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97. [CrossRef]


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.