Research Article

A Provably Secure Authentication and Key Exchange Protocol in Vehicular Ad Hoc Networks

Tsu-Yang Wu, Zhiyuan Lee, Lei Yang, and Chien-Ming Chen

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

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

Received 17 March 2021; Revised 13 May 2021; Accepted 12 June 2021; Published 29 June 2021

Academic Editor: Azeem Irshad

Copyright © 2021 Tsu-Yang Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

While cloud computing and Internet of Things (IoT) technologies have gradually matured, mobile intelligent transportation systems have begun to be widely used. In particular, the application of vehicular ad hoc networks (VANETs) is very convenient for real-time collection and analysis of traffic data. VANETs provide a great convenience for drivers and passengers, making it easier to choose routes. Currently, most research on VANETs obtains data through cloud servers. However, there are few studies on cloud servers obtaining vehicle information through the roadside unit (RSU). In the process of reading traffic information, there will be some private and sensitive information, which may be intercepted or tampered with in untrusted public channels. Therefore, it is necessary to propose a protocol to protect vehicle data during the information reading phase. In this paper, we propose a new provably secure authentication protocol to negotiate a session key before transmitting traffic information. This protocol can complete mutual authentication and generate a session key. Finally, security analysis and performance analysis show that our protocol is secure and efficient.

1. Introduction

Due to social and economic development, motor vehicles are rapidly spreading. At the same time, the rapid increase in the number of vehicles on the road has also made the traffic situation more complicated, and there will be many traffic problems, such as traffic accidents and road congestion. Therefore, researchers apply artificial intelligence [1-4], wireless networks, and sensor technology [5,6] to road vehicle management, so that vehicles can share information and release relevant road information to alleviate traffic problems. This is the vehicular ad hoc network, which consists of vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I) communication. In the VANETs, the vehicle is equipped with an on-board unit (OBU), so the vehicle can be regarded as a mobile network node that can communicate. Therefore, the vehicle can obtain the corresponding road information from the cloud server through the RSU and can also send the relevant information recorded by itself to the cloud server. The main goal of VANET technology is to improve traffic efficiency and increase driving experience. With the popularity and development of VANET, it plays a key role in user travel planning and road safety.

Although VANETs have various obvious benefits, their security and privacy issues [7-9] are still the keys to whether they can be widely used. In VANETs, the network environment is open; attackers can capture various messages transmitted in the network and can forge a legitimate vehicle to send wrong information. The transmission of wrong information will mislead the driver to make the wrong decision, bringing corresponding troubles and even dangers. First, before information transmission, mutual authentication must be performed, and a corresponding session key must be generated for subsequent information transmission. Then the integrity of the message must be verified every time a message is received. In addition, anonymity is indispensable in VANET, because if the vehicle transmits its identity on the network in clear text, the attacker captures the information, and the vehicle can be faked or the vehicle can be tracked.
However, several kinds of research in VANETs mainly focus on how to ensure that vehicles obtain corresponding road information. In other aspects, vehicles can receive current traffic conditions through RSU. Based on the information received, the driver can adjust the driving decision. Because the road conditions are changing, the RSU can actively establish a communication request with the vehicle to obtain the road condition information stored by the vehicle sensor (as shown in Figure 1). Based on our best knowledge, we propose a new provably secure mutual authentication scheme for negotiating session keys before transmitting traffic information in this paper. The main contributions of this paper are summarized as follows:

1. A three-party AKE scheme is proposed, with vehicles, RSU, and cloud servers. RSU actively sends a request, completes mutual authentication with the vehicle through the cloud server, and generates a session key.
2. Due to environmental constraints, the proposed scheme only performs simple operations, such as elliptic curve (ECC), bitwise XOR, and hash functions.
3. We conduct a security analysis of the protocol, including formal analysis, informal analysis, and ProVerif simulation.
4. Finally, the performance of the proposed protocol is evaluated. Compared with the existing methods, we show that our protocol is feasible.

The remainder of this paper is organized as follows. In Section 2, the latest research results of the AKE protocol and related research on security authentication in the VANET environment are reviewed. Section 3 describes our proposed protocol in detail. Then, in Sections 4 and 5, the security analysis and performance analysis of the protocol proposed in Section 3 are carried out. Finally, the article is summarized in Section 6.

2. Related Work

Many researchers have conducted a series of studies on authentication and key exchange protocols in VANETs. However, with the changes of various needs and scenarios, many security issues have emerged in these studies.

First of all, in terms of an authentication protocol, Lamport [10] proposed for the first time password authentication in an insecure channel. Immediately afterward, various two-party authentication schemes were proposed [11, 12]. But, for the VANETs environment, the communication between vehicles can use a two-party authentication scheme, and if the vehicle and the cloud server are authenticated, the two-party authentication will cause transmission delay, because two-party identity authentication is generally used in a single-server environment. In 2001, Li et al. [13] first proposed an authentication scheme in a multiserver environment, but their scheme is inefficient because it takes a lot of time to train neural networks. Later, to complete efficient and secure identity authentication, researchers began to introduce multifactor security. In addition to passwords, security factors such as smart cards and biological information were introduced [14–16]. Recently, Irshad et al. [17] proposed an authentication scheme under a multiserver architecture based on the chaotic mapping. But Wu et al. [18] found that Irshad et al.’s protocol cannot guarantee user anonymity and is vulnerable to attacks by privileged insiders. Therefore, Wu et al. proposed an authentication protocol for distributed cloud environments, claiming that their protocol can resist various known attacks. However, Wu et al. [19] recently proposed an authentication key exchange protocol under a multiserver architecture and found that [18] has multiple security problems, including the inability to provide perfect forward secrecy (PFS) and being vulnerable to privileged internal attacks. Also, in a multiserver environment, in 2017, Truong et al. [20] proposed an ECC-based authentication scheme. Their article discussed that Yeh et al.’s [21] protocol cannot provide mutual authentication and the key agreement phase is incorrect. In 2018, Zhao et al. [22] proposed a secure and efficient authentication protocol based on passwords and smart cards. They claimed that the scheme of Truong et al. could not achieve the security authentication requirements of multisever authentication and could not resist offline password guessing and impersonation attacks. However, Hassaan et al. [23] conducted a security analysis on the scheme proposed by Zhao et al. and found that the scheme is vulnerable to anonymity and traceability issues and is not suitable for a multiserver environment. Then, on this basis, Hassan et al. proposed an improved multiserver authentication scheme.

Currently, there are two research focuses on the VANETs environment; one is efficient authentication, and the other is privacy protection. The former appeared because of the large number of vehicles in the VANETs environment, and data transmission and processing are very challenging. In order to solve this problem, cloud computing began to be applied to the VANETs [24]. In VANETs, cloud computing-based authentication schemes have also begun to be widely proposed [25–29]. These solutions reduce the server-side service response time and improve authentication efficiency. However, due to the number of vehicles involved and management issues, network delays can also be caused. Then cloud computing began to decentralize and fog computing was used to solve the above shortcomings [15, 30–32]. The latter is because, in an open network environment, the private information of vehicle users must be protected. Therefore, the Conditional Privacy Preservation Authentication (CPPA) agreement was proposed [33]. In this protocol, the attacker cannot obtain the true identity of the vehicle user through messages intercepted on the public channel, but a trusted third party can calculate the identity of the vehicle user who sent the message. In 2008, Zhang et al. [34] proposed an identity-based verification scheme and proved that their proposed scheme can practice conditional privacy protection, trusting the authority to retrieve the true identity of the vehicle from any false identity. In 2014, Chuang and Lee [35] proposed the first authentication mechanism using transitive trust relationship. Later, Zhou et al. [36] used elliptic curve cryptography (ECC) to
propose a new mutual authentication scheme based on the mechanism proposed by Chuang and Lee and mentioned in their paper that the scheme of Chang and Lee cannot resist internal attacks. However, Wu et al. [37] found that Zhou et al.’s scheme could not guarantee anonymity and was vulnerable to identity guessing and impersonation attacks. At the same time, they designed a new privacy protection authentication protocol using ECC technology. Some researchers have proposed the use of fog computing for information processing in the VANET environment. In 2019, Ma et al. [30] proposed a new AKE protocol without bilinear pairing. They believed that the proposed protocol is safe and efficient. However, Eftekhari et al. [38] found that the protocol of Ma et al. had security problems, such as internal attacks, known session-specific temporary information attacks, and stolen smart card attacks, and then they proposed a safer and more efficient protocol. In 2017, Mohit et al. [39] proposed a new vehicle communication protocol and believed that their protocol could resist attacks such as stolen smart card attacks and impersonation attacks. However, Yu et al. [40] found that Mohit et al.’s scheme could not provide security attributes such as anonymity and mutual authentication and would suffer impersonation and traceability attacks. Then Yu et al. proposed a new security authentication protocol and proved that their protocol can resist various known attacks. In 2020, Sadri and Rajabzadeh Asaar [41] proved that Yu et al.’s protocol is vulnerable to tracking attacks, impersonation attacks, sensor capture attacks, and so forth and proposed a secure protocol for application in VANETs.

Some studies have begun to design the AKE protocol for the advantages of low latency and high reliability in the 5G environment [42]; and, for some special occasions, blockchain technology [43] is also used to complete the authentication key exchange. Research similar to VANET currently has similar flying ad hoc networks (FANETs). Moreover, this environment is also vulnerable to serious security threats. Due to these security threats, many security protocols have been proposed in this environment [44–47]. Therefore, when studying VANETs, you can refer to some security solutions in FANETs. However, most of the research is carried out on the premise that the vehicle initiates a communication request. So, it is necessary to propose an authentication scheme in which a cloud server or RSU initiates a communication request to the vehicle user to meet the timely update of road condition information.

3. Proposed Scheme

In this section, we introduce in detail a new provably secure mutual authentication scheme used to negotiate session keys before transmitting traffic information. The communication entities in the proposed protocol include vehicle users, roadside units, and cloud servers. For the convenience of reading, the symbols used in the scheme are listed in Table 1. The proposed protocol has five phases, namely, the initialization phase, the vehicle registration phase, the RSU registration phase, the login phase, and the authentication phase.

3.1. Initialization Phase

(1) The cloud server CS selects two large prime numbers \( p \) and \( q \) and then constructs an elliptic curve \( E \) defined about the domain \( Z_q \) for \( q \). The points on \( E \)
Table 1: Notations and descriptions.

| Notations | Description |
|-----------|-------------|
| $V_i$     | The $i_{th}$ vehicle end user |
| $RSU_j$   | The $j_{th}$ roadside unit |
| $CS$      | The cloud server |
| $VID_j$   | $V_j$'s identity |
| $PW_j$    | $V_j$'s password |
| $Bio_i$   | Biometric features of $V_i$ |
| $Gen(\cdot)/Rep(\cdot)$ | Generation/reproduction process of fuzzy extractor |
| $x$       | The secret key of CSP |
| $PVID_i$  | The pseudoidentities of $V_i$ |
| $SK$      | Session key |
| $\mathcal{A}$ | The attacker |
| $h(\cdot)$ | One-way hash function |
| $x\|y$ | Concatenation |
| $x@y$ | The exclusive-or operation with $x$ and $y$ |

form a cyclic additive elliptic curve group $G$, and the generator $P$ of $G$ is obtained.

(2) $CS$ selects two random numbers $x$ and $\alpha$ and computes $\beta = \alpha \cdot P$, where $x$ is the long-term key of the CS, $\alpha$ is the private key, and $\beta$ is the public key.

(3) Finally, CS chooses a one-way hash function $h(\cdot)$.

3.2. Vehicle User Registration Phase. When the vehicle user $V_i$ wants to get the corresponding service, he/she must register through the cloud server $CS$. The main steps are as follows. Figure 2 describes the process of vehicle user registration in detail.

(1) $V_i$ chooses its own $VID_i$ and then sends it to $CS$ through a secret channel.

(2) On receiving $\{VID_i\}$, CS selects $n_i$ and computes $K_v = h(VID_i \| h(x \| n_i))$ and $PVID_i = h(VID_i \| K_v)$. Then, CS saves $\{PVID_i, VID_i, n_i\}$ to memory and securely transmits $\{PVID_i, K_v\}$ to $V_i$.

(3) Finally, $V_i$ computes $(\phi_i, \theta_i) = Gen(Bio_i), HP_i = K_v \oplus h(PW_i \| \phi_i)$, and $Auth_i = h(K_v, \| PVID_i)$ and stores $\{PVID_i, Auth_i, HP_i, \theta_i\}$ into OBU. Among them, $PW_i$ is the $V_i$'s password, and $Bio_i$ is the $V_i$'s biological information.

3.3. RSU Registration Phase. Through the registration phase, $RSU_j$ can obtain the private key, as shown in Figure 3.

(1) $RSU_j$ selects a random number $c_j$ and computes $d_j = c_j \cdot P$ and then sends the identity $RID_j$ and $d_j$ to $CS$ securely.

(2) $CS$ selects the pseudoidentity $PRID_j$ of $RSU_j$ and the random number $k_j$. Then $CS$ computes $y_j = k_j \cdot P + d_j$ and $z_j = k_j + (y_j + PRID_j) \cdot \mathbb{amod} P$, stores $\{PRID_j, RID_j, y_j, k_j\}$ in its database, and finally sends $\{PRID_j, y_j, z_j\}$ to $RSU_j$.

(3) $RSU_j$ computes $x_j = z_j + c_j$ and then verifies whether $x_j \cdot P$ is equal to $z_j \cdot P + c_j \cdot P$. If the verification is passed, the private key distribution is successful. Then $\{PRID_j, y_j, z_j\}$ is stored in $RSU_j$ memory.

3.4. Login Phase. Since the environment proposed by the scheme is to complete mutual authentication and key exchange during vehicle operation, the vehicle user login will be completed in advance. Figure 4 shows the login information of the vehicle user.

3.5. Authentication Phase. The entire authentication phase is initiated by $RSU_j$, which wants to communicate with the running vehicle. The detailed information is shown in Figure 5.

(1) First, $RSU_j$ makes a communication request $Request$ and selects a random number $r_j$ to compute $R_j = r_j \cdot P$. $RSU_j$ sends $Request$ and $R_j$ to $V_j$.

(2) After $V_j$ receives the communication request, it selects a random number $a_i$ and the current timestamp $T_1$ and computes $N_i, M_i, C_1, C_2$ (see equations (1)–(4)). Then it sends $\{N_i, C_1, C_2, T_1\}$ to $RSU_j$.

\[
N_i = a_i \cdot P, \quad (1)
\]

\[
M_i = a_i \cdot \beta, \quad (2)
\]

\[
C_1 = PVID_i + h(M_i \| R_j \| N_i \| T_1), \quad (3)
\]

\[
C_2 = h(PVID_i, \| VID_i \| N_i \| T_1) \quad (4)
\]

(3) $RSU_j$ verifies the validity of the timestamp (by $|T_2 - T_1| < \Delta T$). $RSU_i$ computes $M_j, C_3, C_4$ (as shown in equations (5)–(7)). Finally, $RSU_j$ sends $\{R_j, C_3, C_4, T_2\}$ to $CS$.

\[
M_j = r_j \cdot \beta, \quad (5)
\]

\[
C_3 = PRID_j + h(M_j \| R_j \| T_2), \quad (6)
\]

\[
C_4 = h(\{PRID_j, \| RID_j \| M_j \| T_2\}), \quad (7)
\]

(4) After $CS$ receives the message, it first verifies whether the timestamp is valid (by $|T_2 - T_1| < \Delta T$). If the verification is passed, it computes $M_i = x_j \cdot N_i$ and $PVID_j = C_1 - h(M_i \| R_j \| N_i \| T_1)$. Then it takes out $VID_i$ from the memory through $PVID_i$ and computes $C_2^j = h(PVID_i \| \| VID_i \| N_i \| T_1)$. If $C_2^j$ and $C_2$ are equal, then perform the operation; otherwise, terminate the session. Then, $CS$ computes $M_j = a \cdot R_j$ and $PRID_j = C_3 - h(M_j \| R_j \| T_2)$ and then retrieves $RID_j$ in the database through $PRID_j$. After that, $CS$ computes $C_4^j = h(\{PRID_j, \| RID_j \| M_j \| T_2\})$ and completes the authentication operation. If authenticated, $CS$ selects a random number $b_s$ and a
\[
\begin{array}{|c|c|}
\hline
V_i & CS \\
selects VID_j & \{VID_j\} \\
& \text{picks } n_i \\
& K_v = h(VID_j \| h(x \mid n_i)) \\
& PVID_j = h(VID_j \| K_v) \\
& \text{stores } \{PVID_j, V ID_j, n_i\} \\
\hline
\end{array}
\]

inputs \( Pw_i, Bio_i \)

\[
(\phi_i, \theta_i) = \text{Gen}(Bio_i) \\
H P_i = K_v \oplus h(Pw_i \| \phi_i) \\
Auth_i = h(K_v \| V ID_j)
\]

Stores \( \{PVID_j, Auth_i, H P_i, \theta_i\} \) into OBU

Figure 2: Vehicle user registration phase.

\[
\begin{array}{|c|c|}
\hline
RSU_j & CS \\
c_j, RID_j & \{RID_j, d_j\} \\
d_j = c_j \cdot P \\
& \text{selects } PRID_j, k_j, \\
y_j = k_j \cdot P + d_j \\
z_j = k_j + (y_j + PRID_j) \cdot a \mod p \\
& \text{stores } \{PRID_j, RID_j, y_j, k_j\} \\
\hline
\end{array}
\]

\[
x_j = z_j + c_j \\
\text{checks } x_j \cdot P = z_j \cdot P + c_j \cdot P \\
\text{stores } \{PRID_j, y_j, z_j\}
\]

Figure 3: Registration phase.

\[
\begin{array}{|c|c|}
\hline
V_i & \text{Enters } V ID_j \text{ and } Pw_i \\
\text{inputs biological information } Bio_i & \\
& \phi_i' = \text{Rep}(Bio_i, \theta_i) \\
& K_v' = H P_i \oplus h(Pw_i \| \phi_i') \\
& Auth_i' = h(K_v' \| V ID_j) \\
& \text{Check } Auth_i = Auth_i' \\
\hline
\end{array}
\]

Figure 4: Login phase.

timestamp \( T_3 \) and computes

\[
N_s = b_s \cdot P, \\
C_5 = b_s \cdot (y_j + (y_j + PRID_j) \cdot \beta),
\]

(8)

(9)

\( N_s, C_5, C_6, PVID_{new}, PRID_{new}, C_7 \) (see equations (8)–(13)). Finally, CS updates the values of PRID_j and PVID_j in memory and sends \( \{C_5, C_6, C_7, N_s, T_3\} \) to RSU_j.
selects $a_i$
$N_i = a_i \cdot P$
$M_i = a_i \cdot \beta$
$C_6 = \text{PVID}_i + h(M_i \| R_i \| N_i \| T_1)$
$C_6 = h(\text{PVID}_i \| \text{VID}_i \| N_i \| T_1)$

$[\text{Request}, R_i]$
$[\text{Request}, R_i]$

$C_4 = \text{PRID}_j + h(M_j \| R_j \| T_j)$
$C_4 = h(\text{PRID}_j \| \text{RID}_j \| M_j \| T_j)$

$[T_2 - T_1] < \Delta T$
$M_j = a_j \cdot R_j$
$C_5 = \text{PRID}_j + h(M_j \| R_j \| T_j)$
$C_5 = h(\text{PRID}_j \| \text{RID}_j \| M_j \| T_j)$

$\{N, C_4, C_6, T_1\}$

4. Security Analysis

In this section, we conduct a security analysis of the proposed protocol and use the ROR model and ProVerif tool to complete the formal security analysis [48, 49]; and, through informal security analysis, we verified that the proposed protocol has security features and can resist various known attacks.

4.1. Informal Security Analysis.

This section is an informal security analysis of the proposed protocol. We verify the security attributes and attacks that the proposed protocol needs to have one by one.

4.1.1. Mutual Authentication.

After receiving the authentication request from RSU$_j$, $V_i$ computes $C_6 = \text{PRID}_j + h(\text{VID}_i \| M_j \| T_3)$ and sends it to CS through RSU$_j$. After CS receives the RSU$_j$ message, the computed $C_6$ is equal to $C_5$. If authenticated, RSU$_j$ computes $\text{PVID}_{\text{new}} = h(\text{PVID}_i \| M_j)$ and sends it to $V_i$.

$N_j = b_j \cdot P$

$C_8 = h(\text{PRID}_j \| R_j \| M_j \| T_4)$

$\text{PRID}_{\text{new}} = h(\text{PRID}_j \| M_j)$.

(10) RSU$_j$ also verifies the validity of the timestamp. Then it computes $C_5 = x_j \cdot N_j$ and verifies that $C_5$ is equal to $C_5$. If authenticated, RSU$_j$ computes $\text{PVID}_{\text{new}} = h(\text{PVID}_i \| M_j)$ and $\text{PRID}_{\text{new}} = h(\text{PRID}_j \| M_j)$.

$[\text{Request}, R_i]$

$[\text{Request}, R_i]$

$[T_2 - T_1] < \Delta T$
$M_j = a_j \cdot R_j$
$C_5 = \text{PRID}_j + h(M_j \| R_j \| T_j)$
$C_5 = h(\text{PRID}_j \| \text{RID}_j \| M_j \| T_j)$

$\{R_i, C_4, C_6, T_1\}$

$[T_2 - T_1] < \Delta T$
$C_4 = x_j \cdot N_j$
$C_4 = x_j \cdot N_j$
$C_4 = x_j \cdot N_j$
$C_4 = x_j \cdot N_j$

4.1.2. Man-in-the-Middle Attacks.

By intercepting the information in the public channel, $A$ may launch man-in-the-middle attacks. If authenticating, $V_i$ computes $C_2 = h(\text{PVID}_i \| \text{VID}_i \| N_i \| T_1)$ and sends it to RSU$_j$. After RSU$_j$ receives the request, the computed $C_2$ contains the parameters $\{\text{PVID}, \text{VID}\}$. Only legitimate users can generate correct $C_2$, so that RSU$_j$ can verify the identity of the user and the legitimacy of the information by verifying whether $C_2$ is equal to $C_2$; that is, RSU$_j$ authenticates $V_i$. Similarly, the server computes $C_4$, RSU$_j$ computes $C_6$, and $V_i$ computes $C_8$, respectively, indicating that $V_i$ has authenticated RSU$_j$, RSU$_j$ has authenticated CS, and $V_i$ has authenticated CS. In summary, $V_i$ and RSU$_j$ can perform mutual authentication in the protocol.

4.1.3. Replay Attacks.

By intercepting the information in the public channel, $V_i$ may replay the authenticating message. If authenticating, $V_i$ computes $C_2 = h(\text{PVID}_i \| \text{VID}_i \| N_i \| T_1)$ and sends it to RSU$_j$. After RSU$_j$ receives the request, the computed $C_2$ contains the parameters $\{\text{PVID}, \text{VID}\}$. Only legitimate users can generate correct $C_2$, so that RSU$_j$ can verify the identity of the user and the legitimacy of the information by verifying whether $C_2$ is equal to $C_2$; that is, RSU$_j$ authenticates $V_i$. Similarly, the server computes $C_4$, RSU$_j$ computes $C_6$, and $V_i$ computes $C_8$, respectively, indicating that $V_i$ has authenticated RSU$_j$, RSU$_j$ has authenticated CS, and $V_i$ has authenticated CS. In summary, $V_i$ and RSU$_j$ can perform mutual authentication in the protocol.

4.1.4. Man-in-the-Middle Attacks. By intercepting the information in the public channel, $A$ may launch man-in-the-
middle attacks. But after CS receives the message, it needs to verify $C_3^j = h(\text{VID}_j \parallel N_i \parallel T_i)$ and $C_4^j = h(\text{PRID}_j \parallel \text{RID}_j \parallel M_j \parallel T_j)$ to authenticate the sender. Suppose that $\mathcal{A}$ tries to tamper with the information sent to $\text{RSU}_j$, he needs to generate a new authentication information $C_3$, but he cannot obtain the parameters $x_j$, $b_j$, and so forth. This means that $\mathcal{A}$ cannot complete the verification after tampering with the information. Similarly, when $\mathcal{A}$ tampers with the information sent to $V_j$ and CS, he could not complete the relevant authentication. This shows that the protocol can resist man-in-the-middle attacks.

4.1.3. Replay Attacks. In the protocol, when a new round of authentication is performed, new random numbers $r_j$, $a_i$, and $b_i$ will be generated; and every time the authentication is completed, the values stored in the memory such as $\text{VID}_i$ and $\text{PRID}_i$ will be updated. The random number and the updated $\text{VID}_i$ are used when generating the session key. Therefore, when $\mathcal{A}$ resends the previous message, new random numbers and related parameters updated in the memory have been generated, and he cannot pass the verification and cannot compute the session key. Therefore, the proposed protocol can resist replay attacks.

4.1.4. Known Session-Specific Temporary Information Attacks. Under the CK attack model [50], $\mathcal{A}$ can obtain the random number $a_i$ or $r_j$ generated during the authentication phase. Assuming that $\mathcal{A}$ obtains the random number $a_i$ generated by $V_j$ then $N_i$, $M_i$, and $\text{VID}_i$ can be calculated. However, since $\mathcal{A}$ cannot obtain $\text{VID}_i$ and $\text{PRID}_i$, he still cannot compute the session key $SK$; and when $\mathcal{A}$ tries to use a random number to perform a man-in-the-middle attack or an impersonation attack, he cannot complete the verification by recalculating $C_2$. Therefore, the proposed protocol can resist known session-specific temporary information attacks.

4.1.5. Perfect Forward Secrecy. This security feature requires that the leakage of the long-term key does not reveal the previously generated session key. $SK = h(\text{VID}_{\text{inew}} \parallel \text{PRID}_j \parallel a_i \cdot R_j) = h(\text{VID}_{\text{inew}} \parallel \text{PRID}_j \parallel r_j \cdot N_j)$ in the scheme. That is, the long-term key $x$ of CS is not used in the calculation of the session key. Since the private key $a_i$ of CS does not change after each authentication, it is assumed that $\mathcal{A}$ can get $a_i$. Then $\mathcal{A}$ can compute $M_i = a_i \cdot N_i$ and $M_j = a_i \cdot R_j$; that is, $\text{VID}_i = C_1 = h(M_i \parallel N_i \parallel T_i)$ and $\text{PRID}_j = C_1 = h(M_i \parallel R_j \parallel T_j)$ and the updated $\text{VID}_{\text{inew}} = h(\text{VID}_i \parallel M_i)$ and $\text{PRID}_{\text{inew}} = h(\text{PRID}_j \parallel M_j)$. However, $\mathcal{A}$ cannot obtain the random number $a_i$ or $r_j$ needed to compute $SK$, so there is no way to compute $SK$; that is, the proposed protocol can provide perfect forward secrecy.

4.1.6. Internal Attacks. Assuming that $\mathcal{A}$ is a CS internal staff, he can easily obtain the information transmitted during the registration phase, including $\{\text{VID}_i\}$, $\{\text{VID}_j, K_j\}$, $\{\text{PRID}_j, y_j, z_j\}$, and $\{\text{PRID}_j, y_j, z_j\}$. However, $\mathcal{A}$ cannot compute $a_i$ and $r_j$ from this information. Therefore, the proposed protocol can resist internal attacks.

4.1.7. User Anonymity and Untraceability. During the authentication process, $\text{VID}_i$ is used to compute $C_2$ and $\mathcal{A}$ cannot obtain $\text{VID}_i$ to guess $\text{VID}_j$. So, the scheme can guarantee anonymity. At the same time, due to the use of random numbers and the update of the pseudoidentity after each authentication, it is also ensured that $\mathcal{A}$ cannot confirm the user’s identity by tracing a specific piece of information. Therefore, the protocol satisfies anonymity and untraceability.

4.1.8. Three-Factor Secrecy. The proposed protocol uses passwords, biological information, and storage devices (OBU) for security encryption, so it is a three-factor authentication protocol. For this type of protocol, it is assumed that the extreme case is that $\mathcal{A}$ can obtain two of the three factors and can launch an attack on the protocol.

Assume that $\mathcal{A}$ obtains $\text{VID}_j$, $\text{PRID}_j$, and $\text{Bio}_j$. It is necessary to compute $\text{Auth}_i$ when logging in, where $K_i = HP_i = h(\text{PRID}_j \parallel \text{Bio}_j)$, but $HP_i$ is stored in OBU. In other words, $\mathcal{A}$ cannot complete the login operation. The proposed protocol is safe in this situation. Assume that $\mathcal{A}$ obtains $\text{VID}_j$, $\text{PRID}_j$, and $\text{Bio}_j$. Since $\text{Bio}_j$ cannot be computed through $\text{Bio}_j$, $\mathcal{A}$ cannot compute $K_i$ and $\text{Auth}$ and cannot complete login verification. That is, the protocol is safe in this situation. Similarly, when $\mathcal{A}$ knows $\text{Bio}_i$ and OBU, there is no way to compute $\text{Auth}$, because there is no password and identity. Therefore, the protocol is safe in the three situations, and the proposed protocol satisfies the three-factor security characteristics.

4.1.9. No Key Control. In this protocol, the session key $SK$ can only be generated through negotiation between $V_j$ and $\text{RSU}_j$; that is, a single entity cannot generate $SK$ by itself. When computing $SK$, $V_j$ needs to know $\text{PRID}_j$ and $R_j$ generated by $\text{RSU}_j$. In the same way, $\text{RSU}_j$ needs to negotiate to obtain $\text{VID}_{\text{inew}}$ and $N_j$ during the calculation to compute $SK$. Therefore, the proposed protocol is satisfied with no key control property.

4.2. Formal Security Analysis Based on Random Oracle Model. In this section, a random oracle model (ROR model) is used to formally prove the security of our proposed protocol. This analysis model was proposed by Canetti et al. [51]. By launching different rounds of Games, the ROR model can compute the probability of $\mathcal{A}$ successfully guessing the SK in various situations and thus judge the security of the protocol. Assume that $I_{x_j}^y$, $I_{\text{RSU}_j}^y$, and $I_{\text{CS}}^y$, respectively, represent the $x$th communication of $V_j$, the $y$th communication of $\text{RSU}_j$, and the $z$th communication of CS. $\mathcal{A}$ can initiate the following query, where $O = \{I_{x_j}^y, I_{\text{RSU}_j}^y, I_{\text{CS}}^y\}$:

(i) Execute $(O)$: through this query, $\mathcal{A}$ can eavesdrop on the message transmitted on the public channel.
(ii) Hash (string): $\mathcal{A}$ executes the query and can get the hash value of the input parameter string.
Adv ECDLP

(iii) Send \((O, M)\): \(A\) executes the query, sends a message \(M\) to \(O\), and can receive the corresponding response.

(iv) Reveal \((O)\): \(A\) executes this query to obtain the return result of current session key SK generated by \(M\).

(v) Corrupt \((O)\): by executing the query, \(A\) can obtain some secret values, such as long-term private keys and temporary information.

(vi) Test \((O)\): \(A\) executes the query and judges the correctness of the session key by flipping coin \(C\). If the result is \(C = 1\), \(A\) will receive the correct session key returned; if the result is \(C = 0\), \(A\) will receive a random string.

\[\text{Definition 1. (elliptic curve discrete logarithm problem (ECDLP))} \]

Our proposed protocol uses elliptic curve cryptography (ECC). Here, we describe the computational difficulties and assumptions of ECC. Suppose that \(C\) is an elliptic curve generation group. At the same time, given points \(P\) and \(a \cdot P\), where \(P\) belongs to \(C\) and \(a\) belongs to \(F_p\), it is computationally infeasible to obtain \(a\). In polynomial time, the probability that \(A\) solves this problem is defined as follows:

\[\text{Adv}_A^{ECDLP}(\xi) = Pr[A(P, aP) = a; a \in F_p, P \in \xi].\]

For a sufficiently small \(\eta\), we have \(\text{Adv}_A^{ECDLP}(\xi) < \eta\).

\[\text{Theorem 1. If } A \text{ attempts to initiate some queries in polynomial time, then the advantage that he can break through the proposed protocol } P \text{ is as follows: } \text{Adv}_A^{ECDLP}(\xi) = (q_{hash}^2/2^l)\text{Adv}_A^{ECDLP}(\xi) + 2\text{max}(C', q_{send}^2/2^l) + 2q_{send}\text{Adv}_A^{ECDLP}(\xi) + (q_{send}^2 + q_{exec}^2)/p + (q_{hash}^2/2^l) + (q_{send}/2^{l-1}),\]

where \(q_{hash}\) represents the number of times to execute \(\text{Hash(string)}\) queries, \(q_{send}\) represents the number of times to execute \(\text{Send(O,M)}\) queries, \(q_{exec}\) represents the number of times to execute \(\text{Execute(O)}\) queries, and \(C', l\) and \(s\) are constants in Zipf's law [52].

\[\text{Proof. We use the game sequence } GM_0, GM_1, GM_2, GM_3, GM_4, GM_5, GM_6 \text{ to verify the above theorem. } \]

\[\text{Succ}_{GM_0}^{A}(\xi) \text{ represents the probability of } A \text{'s success in game } GM_0. \text{ Finally, using the Test query to determine } \text{Succ}_{GM_2}^{A}(\xi), \text{ the specific description is as follows:} \]

(i) \(\text{Game } GM_0\): \(GM_0\) represents a real attack, and \(A\) did not initiate any query at this time. Therefore, in \(GM_0\), the probability of \(A\) cracking \(P\) is \(\text{Adv}_A^{GM_0}(\xi) = 2\text{Pr[GM_0(\xi)]} - 1\).

(ii) \(\text{Game } GM_1\): \(GM_1\) adds Execute query on the basis of \(GM_0\), and there is no difference in the others. So, \(\text{Pr[GM_1(\xi)]} = \text{Pr[GM_0(\xi)]}\).

(iii) \(\text{Game } GM_2\): \(GM_2\) adds Send query on the basis of \(GM_1\). According to Zipf's law, we get \(\text{Pr[GM_2(\xi)]} = \text{Pr[GM_0(\xi)]} \leq q_{send}/2^l\).

(iv) \(\text{Game } GM_3\): \(GM_3\) adds Hash query on the basis of \(GM_2\). According to the birthday paradox, we can get the maximum probability of hash collision as \(q_{hash}^2/2^l\); the maximum probability of collision in the transmitted text is \((q_{hash} + q_{exec})^2/2p\); and so \(\text{Pr[GM_3(\xi)]} \leq q_{send}^2/2p + q_{hash}^2/2^l\).

(v) \(\text{Game } GM_4\): In this game, we consider the security of the session key. Here, to divide the discussion into two situations. The first is to obtain a long-term private key to verify perfect forward secrecy; the second is to provide temporary information leakage to verify whether the known session-specific temporary information attacks can be resisted.

\[\text{Perfect forward secrecy: } A \text{ uses } \text{Corrupt(}I_y\text{)} \text{ to try to get the private key } x_j \text{ of RSU}, \text{ or uses } \text{Corrupt(}I_y\text{)} \text{ or } \text{Corrupt(}I_x\text{)} \text{ to try to get a certain secret value in the registration phase} \]

\[\text{Known session-specific temporary information attacks: } A \text{ uses } \text{Corrupt(}I_y\text{)} \text{ or } \text{Corrupt(}I_x\text{)} \text{ or } \text{Corrupt(}I_y\text{)} \text{ to try to obtain temporary information of one party} \]

In both cases, ECDLP needs to be solved to compute the session key SK. For \(\text{SK} = h(\text{VID}_{\text{new}} || \text{PRID}_j || a_i \cdot R_j)\), in the first case, even if \(M_j\) and \(\text{VID}_i\) are calculated by \(x_j\), the random number \(r_j\) is unknown. While getting \(a_i\) through \(\text{Corrupt(}I_y\text{)}\), \(A\) cannot get \(\text{VID}_i\), \(\text{VID}_i\). In the second case, even if \(a_i \cdot R_j\) is calculated through \(a_i\), the long-term private key is unknown. Similarly, for the second formula \(\text{SK} = h(\text{VID}_{\text{new}} || \text{PRID}_j || r_j \cdot N_j)\) also holds, \(\text{Pr[GM_4(\xi)]} = \text{Pr[GM_3(\xi)]} \leq q_{send}^{GM_3} \text{Adv}_A^{ECDLP}(\xi)\).

(i) \(\text{Game } GM_5\): \(GM_5\) uses \(\text{Corrupt(}I_x\text{)}\) to query; \(A\) can get the information \((\text{VID}_i, \text{Auth}_h, \text{HP}_i, \theta)\) in OBU. The user uses the password and biological information to register. \(A\) wants to guess \(\theta\), but the possibility of guessing the biological characteristics is \((1/2)^l\), which can be almost ignored. Using Zipf's law, we can get \(\text{Pr[GM_5(\xi)]} = \text{Pr[GM_4(\xi)]} \leq \max(C', q_{send}^2/2^l)\).

(ii) \(\text{Game } GM_6\): the purpose of this game is to verify forgery attacks. In \(GM_6\), \(A\) issues \(h(\text{VID}_{\text{new}} || \text{PRID}_j || r_j \cdot N_j)\) or \(h(\text{VID}_{\text{new}} || \text{PRID}_j || a_i \cdot R_j)\) query, the game is terminated. At this point, the probability of \(A\) guessing SK is \(\text{Pr[GM_6(\xi)]} = \text{Pr[GM_5(\xi)]} \leq q_{send}^{GM_5} \text{Adv}_A^{ECDLP}(\xi)\). Because the probability of success and unsuccess of \(GM_6\) is half, \(\text{Pr[GM_6(\xi)]} = 1/2\).

In summary, we can get the following conclusions:
4.3. ProVerif Security Analysis. ProVerif [53] is a formal cryptographic protocol security verification tool proposed by Bruno Blanchet in 2001 and developed using the Prolog language. The tool is based on the DY model and can handle basic cryptographic operations such as symmetric encryption and decryption, public-key encryption and decryption, hash operations, and XOR operations. Through the use of code to achieve the registration and authentication phases of vehicle users, RSUs, and cloud servers, a protocol simulation experiment is created in this section. The following is the whole process:

(1) The definition of the channel is ch and sch. The former is a common channel used in the login and authentication phases, and the latter is a secure channel used in the registration phase. SK and SK are the session keys generated by OBU and S. The subsequent definitions are string concatenation operations, XOR operations, hash functions, and fuzzy extractor functions. Next is to use some queries to verify the security requirements. The entire definition is shown in Figure 6.

(2) The process of Vi is shown in Figure 7.

(3) The process of RSUj is shown in Figure 8.

(4) The process of CS is shown in Figure 9.

(5) In Figure 10, we show the results of the verification. We use VehicleStarted(), VehicleAuthed(), ServerAcVehicle(), ServerAcRSU(), RSUAcServer(), and VehicleAcRSU() to declare the beginning and the end of the agreement and whether the mutual authentication between the vehicle user, RSU, and CS is correct. The verification result shows that the session key we established has withstood the attack, and the mutual authentication is successful and correct. The protocol proposed in this chapter has passed the security verification of ProVerif.

5. Security and Performance Comparisons

This section will analyze the performance of the proposed protocol and verify the performance of the protocol by comparing its security, computing consumption, and communication consumption among similar protocols.

5.1. Security Comparisons. In this section, we compare the security of the proposed protocol with Ma et al.’s scheme [30], Jia et al.’s scheme [31], Eftekhar et al.’s scheme [38], and Liu et al.’s scheme [54]. The details are shown in Table 2. According to the informal security analysis above, it can be seen that the current common network attacks mainly include A1: mutual authentication; A2: man-in-the-middle attacks, A3: replay attacks, A4: known session-specific temporary information attacks, A5: perfect forward secrecy, A6: internal attacks, A7: user anonymity, A8: three-factor secrecy, A9: no key control, and A10: impersonation attacks. Yes means that it can resist this attack or has this security feature.

5.2. Performance Comparisons. In the performance analysis of the AKE protocol, the computation cost is an important part to be considered. In the VANETs
environment, due to the mobility of vehicles, the required computational time needs to be less, which reduces the time required for key establishment and makes the proposed protocol more practical. The experimental environment we used here is shown in Table 3 to test the time-consuming performance of different encryption and decryption algorithms. The algorithm was run 30 times on the device to find the average value. The results are shown in Table 4. We found that the time of the fuzzy extraction function is similar to that of the hash function during the experiment, so we use the fuzzy extraction function as a hash function.

Compared with other phases, in order to ensure the security of the session key, the authentication phase will be executed multiple times, so the calculation cost in this section only considers the calculation performed in the authentication phase. The comparison is shown in Table 5. Substitute the execution data in Table 4 to get the computation cost histogram in Figure 11.

Next, we analyze the communication consumption of the proposed protocol and compare it with related protocols. We use the number of bits specified in [11]. For example, the point of the ECC is 320 bits, the hash function is set to 256 bits, the length of the identity information is 64 bits, and the length of the random number and timestamp is 32 bits.

The protocol we propose has four transmission rounds in the authentication phase, and the transmitted information is \( R_j, N_j, C_1, C_2, C_3, C_4, C_5, N_s, T_3, T_4 \). It contains 4 ECC points, 8 hash function outputs, and 4 timestamps' information. That is, a total of 3456 bits of information are transmitted.

The protocol of Liu et al. transmits 4 rounds, and the transmitted information is \( \{\text{AID}_i, \text{AID}_j, \text{MAC}, \text{TS}_r, \text{Pr}_r, p_{pub}, C, \text{AID}_j, \text{MAC}, \text{TS}_r, \text{Pr}_r, \{\text{AID}_m, \text{TD}_m, E_{\text{sym}}(q^{\alpha r}), \text{TS}_r, \sigma, M_m\} \), including 3 hash outputs, 4 identification information, 4 timestamps' information, 3 ECC points, and 3 symmetric encryptions' information (calculated according
Figure 7: The process of $V_i$.

```haskell
(* -------------------Vehicle's process------------------- *)
let ProcessVehicle=
    new VIDi : bitstring; (* the Vehicle's ID *)
    out(sch, (VIDi));
    in(sch, (xVIDi:bitstring,xKv:bitstring));
    new PWi : bitstring; (* the Vehicle's password *)
    new Bioi : bitstring; (* the Vehicle's biometric *)
    let (a: bitstring, b: bitstring)=Gen(Bioi) in
    let HPi=xor(xKv,h(cons(PWi,a))) in
    let Authi=h(cons(xKv,VIDi)) in
    !
    (event VehicleStarted();
    let a'=Rep(Bioi,b) in
    let Kv'=xor(HPi,h(cons(PWi,a')))) in
    let Authi'=h(cons(Kv',VIDi)) in
    if Authi'=Authi then
    in(ch, (xRequest:bitstring,xRj:bitstring));
    new ai:bitstring;
    new xFRIDj:bitstring;
    new T1:bitstring;
    let Ni=mult(ai,P) in
    let M1=mult(ai,B) in
    let C1=mul(xVIDi,h(cons(cons(Mi,xRj),Mi),T1)) in
    let C2=h(cons(cons(cons(xVIDi,VIDi),Ni),T1)) in
    out(ch, (Ni,C1,C2,T1)); (*--------authentication-------- *)
    event VehicleAuthed();
    in(ch, (xC:bitstring,xC8:bitstring,xT3:bitstring,xT4:bitstring));
    let FRIDj'=sub(xC8,h(cons(cons(VIDi,Mi),xT3))) in
    let C8'='h(cons(cons(FRIDj'),xRj),Mi)) in
    if C8'=xC8 then event VehicleAcRSU();
    let PVIDinew=h(cons(xVIDi,Mi)) in
    let SKv=h(cons(cons(PVIDinew,FRIDj'),mult(ai,xRj))) in
    0).
```

Figure 8: The process of $RSU_j$.

```haskell
(* -----------RSU's process------------- *)
let ProcessRSU=
    new c:j:bitstring;
    new RIDj:bitstring;
    let dj=mult(c,j,P) in
    out(sch, (RIDj,dj));
    in(sch, (yFRIDj:bitstring,yy:bitstring,yz:bitstring));
    let xj=mul(yj,cj) in
    let xj2=mul(mult(yj,P),mult(cj,P)) in
    if xj=xj2 then
    !
    new Request:bitstring;
    new rj:bitstring;
    new T2:bitstring;
    new T4:bitstring;
    let Rj=mult(rj,P) in
    out(sch, (Request,Rj));
    in(ch, (yNi:bitstring,yC1:bitstring,yC2:bitstring,yT1:bitstring));
    let Mj=mult(rj,B) in
    let C3=mul(yFRIDj,h(cons(cons(Mj,Rj),T2))); in
    let C4='h(cons(cons(cons(yFRIDj,RIDj),T3),T2)) in
    out(ch, (Rj,C3,C4,T2));
    in(ch, (yC5:bitstring,yC6:bitstring,yC7:bitstring,yN:bitstring,yT3:bitstring));
    let C5'=mul(xj,Ws) in
    if C5'=yC5 then event RSUAcServer();
    new yFRIDj:bitstring;
    let PVIDinew=sub(yC7,h(cons(cons(yFRIDj,cons(Mj,yT3))))) in
    let C8=h(cons(cons(cons(yFRIDj,Rj),T4))) in
    let FRIDjnew=h(cons(cons(yFRIDj,Mj)) in
    let SKv=h(cons(cons(PVIDinew,yFRIDj),mult(rj,N1))) in
    out (ch, (yC6,C8,yT3,T4));
    0).
```
Therefore, a total of 2496 bits of information are transmitted. The protocol of Jia et al. transmits 4 rounds, and the transmitted information is \{A, PID_i, N_i, T_u\}, \{A, B, PID_j, N_j, L_j, T_v, T_f\}, \{C, Auth_i, Auth_j, T_c\}, and \{B, C, Auth_j, T_c\}. It includes 6 ECC points, 9 hash function outputs, and 5 timestamps' information. Therefore, a total of 4384 bits of information are transmitted.

The protocol of Ma et al. transmits 4 rounds, and the transmitted information is \{AID_i, T_u, R_1, \alpha, AID_j, T_v, R_2, R_3, R_4, R_5, T_CS, y', T_f\}. It contains 7 ECC points, 3 hash function outputs, and 4 timestamps' information. A total of 3904 bits of information are transmitted.

The protocol of Eftekhari et al. transmits 4 rounds, and the transmitted information is \{Rid_{DS}, X_{VE}, y_{VE}, h_{CS}, T_r\}, \{Rid_{FS}, Rid_{DS}, X_{FS}, y_{VE}, h_{CS}, T_r\}, \{mRid^{DS}_{CS}, mRid^{FS}_{CS}, h_{CS}, h_{FS}, X_{CS}, X_{FS}\}, and \{mRid^{DS}_{CS}, mRid^{FS}_{CS}, h_{CS}, h_{FS}, X_{CS}, X_{FS}, T_r\}. It contains 6 ECC points, 14 hash function outputs, and 2 timestamps' information. A total of 5568 bits of information are transmitted. The comparison of communication consumption is shown in Table 6. In order to see the comparison effect more clearly, we have generated Figure 12.

Combined with Tables 2, 5, and 6, we discussed the results of the performance analysis. The protocol of Eftekhari et al. has no obvious security vulnerabilities, and the computation cost is similar to that of the protocol we proposed; the main computation cost gap is on the server side. Because the server has strong computing power, it has little effect on the overall computation cost; and, from Table 6 we can see that the communication cost of Eftekhari et al.'s protocol is much higher than that of the proposed protocol. Also, the proposed protocol is similar to Jia et al.'s protocol in terms of computation cost, but Jia et al.'s protocol has security vulnerabilities. The communication cost of all the schemes participating in the comparison is slightly higher than that of the protocol of Liu et al. It can be seen from Figure 11 that the computation cost of the protocol of Liu et al. is the highest, and the security...
The computation cost and communication cost of Ma et al.'s protocol are relatively average, but both are slightly higher than those of our proposed protocol, and their protocol is vulnerable to known session-specific temporary information attacks and internal attacks and cannot guarantee user anonymity. In general, it is more reasonable for the proposed protocol to combine security, computation cost, and communication cost analysis.

Figure 10: Verification result.

### Table 2: Security comparison.

| Attack methods | Liu et al. | Ma et al. | Jia et al. | Eftekhari et al. | Our scheme |
|----------------|------------|-----------|------------|-------------------|------------|
| A1             | Yes        | Yes       | No         | Yes               | Yes        |
| A2             | Yes        | Yes       | —          | —                 | Yes        |
| A3             | —          | Yes       | Yes        | Yes               | Yes        |
| A4             | No         | No        | No         | Yes               | Yes        |
| A5             | No         | Yes       | Yes        | Yes               | Yes        |
| A6             | No         | No        | Yes        | Yes               | Yes        |
| A7             | No         | No        | Yes        | Yes               | Yes        |
| A8             | —          | No        | Yes        | —                 | Yes        |
| A9             | Yes        | Yes       | Yes        | Yes               | Yes        |
| A10            | Yes        | Yes       | Yes        | Yes               | Yes        |

### Table 3: Experimental environment.

| Denomination            | Description            |
|-------------------------|------------------------|
| Hardware equipment      | Laptop                 |
| Processor               | AMD Ryzen 5 4600H      |
| Running memory          | 16 GB                  |
| System                  | Windows 10             |
| Software                | IntelliJ IDEA 2019.3   |
| Cryptography library    | JPBC-2.0.0             |

Performance is very poor. The computation cost and communication cost of Ma et al.'s protocol are relatively average, but both are slightly higher than those of our proposed protocol, and their protocol is vulnerable to known session-specific temporary information attacks and internal attacks and cannot guarantee user anonymity. In general, it is more reasonable for the proposed protocol to combine security, computation cost, and communication cost analysis.
Table 4: The computational cost of complex operations.

| Operation | Definition                  | Execution time (ms) |
|-----------|----------------------------|---------------------|
| $T_{\text{pm}}$ | Elliptic curve scalar point multiplication | 8.8 |
| $T_{\text{pa}}$ | Elliptic curve scalar point addition | 0.057 |
| $T_h$ | Hash function | 0.0058 |
| $T_{\text{bp}}$ | Bilinear pairing | 11.43 |
| $T_{\text{h2p}}$ | String to point hash operation | 26.1 |
| $T_{\text{sc}}$ | Symmetric encryption | 18.37 |

Table 5: Computation cost comparison.

| Scheme          | $V_i$             | $\text{RSU}_i$ | CS | Total                          |
|-----------------|-------------------|----------------|----|--------------------------------|
| Liu et al.      | $T_{\text{pm}} + 2T_{\text{h2p}} + T_{\text{bp}} + T_{\text{sc}}$ | $T_{\text{pm}} + 2T_{\text{h2p}} + T_{\text{bp}} + 2T_{\text{sc}}$ | $T_{\text{pm}} + 2T_{\text{h2p}} + T_{\text{sc}}$ | $3T_{\text{pm}} + 6T_{\text{h2p}} + 2T_{\text{bp}} + 4T_{\text{sc}}$ |
| Jia et al.      | $2T_{\text{pm}} + 5T_h + T_{\text{bp}}$ | $2T_{\text{pm}} + 4T_h + T_{\text{bp}}$ | $3T_{\text{pm}} + 11T_h + T_{\text{bp}}$ | $7T_{\text{pm}} + 21T_h + 3T_{\text{bp}}$ |
| Ma et al.       | $3T_{\text{pm}} + 3T_h$ | $4T_{\text{pm}} + 3T_h$ | $6T_{\text{pm}} + 9T_h$ | $13T_h + 15T_h$ |
| Eftekhari et al.| $3T_{\text{pm}} + 13T_h + T_{\text{pa}}$ | $2T_{\text{pm}} + 15T_h + T_{\text{pa}}$ | $3T_{\text{pm}} + 17T_h + 2T_{\text{pa}}$ | $8T_{\text{pm}} + 45T_h + 4T_{\text{pa}}$ |
| Ours            | $2T_{\text{pm}} + 3T_h$ | $3T_{\text{pm}} + 5T_h$ | $4T_{\text{pm}} + 8T_h$ | $9T_{\text{pm}} + 18T_h$ |

Figure 11: Computation cost histogram.

Table 6: Comparison of communication and massage rounds.

| Scheme          | Communication cost (bits) | Massage rounds |
|-----------------|--------------------------|----------------|
| Liu et al.      | 2496                     | 4              |
| Jia et al.      | 4384                     | 4              |
| Ma et al.       | 3904                     | 4              |
| Eftekhari et al.| 5568                     | 4              |
| Our scheme      | 3456                     | 4              |
6. Conclusion

Based on ECC, this paper designs a new provably safe AKE scheme before transmitting road condition information. We first reviewed the research status of AKE protocol in the VANET environment and found that it is necessary to propose a scheme to protect vehicle data in the information reading phase. We conducted an informal security analysis of the proposed protocol from mutual authentication, anonymity, perfect forward secrecy, man-in-the-middle attacks, internal attacks, and so forth and passed strict formal security analyses, such as the ROR model and ProVerif security verification tools, indicating that the proposed protocol is secure. Through the comparison of security and performance, the proposed protocol is secure, more effective, and more reasonable than the existing protocol. The application of authentication and key exchange in the VANETs environment is the general trend of the development of the VANETs. With the continuous development of the VANETs, subsequent application scenarios are also diverse, such as social Internet of Vehicles, which involve more user privacy information, and this topic will have great research value and research space in the future. Therefore, the communication security of the VANETs environment must also be a key research topic for scholars.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] Z. Meng, J.-S. Pan, and K.-K. Tseng, “PaDE: an enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization,” Knowledge-Based Systems, vol. 168, pp. 80–99, 2019.

[2] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, “An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems,” Information Sciences, vol. 561, pp. 304–325, 2021.

[3] X. Xue, X. Wu, C. Jiang, G. Mao, and H. Zhu, “Integrating sensor ontologies with global and local alignment extractions,” Wireless Communications and Mobile Computing, vol. 2021, Article ID 6625184, 10 pages, 2021.

[4] X. Xue, C. Yang, C. Jiang, P. W. Tsai, G. Mao, and H. Zhu, “Optimizing ontology alignment through linkage learning on entity correspondences,” Complexity, vol. 2021, Article ID 5574732, 12 pages, 2021.

[5] H. Xiong, Y. Wu, C. Jin, and S. Kumari, “Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIOT,” IEEE Internet of Things Journal, vol. 7, no. 12, Article ID 11713, 2020.

[6] H. Xiong, Y. Zhao, Y. Hou et al., “Heterogeneous signcryption with equality test for IIoT environment,” IEEE Internet of Things Journal, 2020.

[7] J. S. Pan, X. X. Sun, S. C. Chu, A. Abraham, and B. Yan, “Digital watermarking with improved SMS applied for QR code,” Engineering Applications of Artificial Intelligence, vol. 97, Article ID 104049, 2021.

[8] J. M.-T. Wu, G. Srivastava, A. Jolfaei, P. Fournier-Viger, and J. C.-W. Lin, “Hiding sensitive information in eHealth datasets,” Future Generation Computer Systems, vol. 117, pp. 169–180, 2021.

[9] J. M. T. Wu, G. Srivastava, U. Yun, S. Tayeb, and J. C. W. Lin, “An evolutionary computation-based privacy-preserving data mining model under a multithreshold constraint,” Transactions on Emerging Telecommunications Technologies, vol. 32, no. 3, Article ID e4209, 2021.

[10] L. Lamport, “Password authentication with insecure communication,” Communications of the ACM, vol. 24, no. 11, pp. 770–772, 1981.

[11] T. Y. Wu, Y. Q. Lee, C. M. Chen, Y. Tian, and N. A. Al-Nabhan, “An enhanced pairing-based authentication scheme for smart grid communications,” Journal of Ambient Intelligence and Humanized Computing, 2021.

[12] J. C. Hsu, Y. S. Iheng, S. M. Mizanur Rahman, and R. Tso, “Password-based authenticated key exchange from lattices for client server model,” Journal of Computer Security and Data Forensics, vol. 1, no. 1, pp. 1–17, 2021.

[13] L. H. Li-Hua Li, L. C. Luon-Chang Lin, and M. S. Min-Shiang Hwang, “A remote password authentication scheme for multiserver architecture using neural networks,” IEEE Transactions on Neural Networks, vol. 12, no. 6, pp. 1498–1504, 2001.

[14] C. C. Chang and J. S. Lee, “An efficient and secure multi-server password authentication scheme using smart cards,” in Proceedings of the 2004 International Conference on Cyberworlds, pp. 417–422, Tokyo, Japan, November 2004.

[15] Y. Wang, Y. Liu, H. Ma, Q. Ma, and Q. Ding, “The research of identity authentication based on multiple biometrics fusion in complex interactive environment,” Journal of Network Intelligence, vol. 4, no. 4, pp. 124–139, 2019.

[16] T. Y. Wu, Z. Lee, L. Yang, J. N. Luo, and R. Tso, “Provably secure authentication key exchange scheme using fog nodes in vehicular ad hoc networks,” The Journal of Supercomputing, vol. 77, pp. 6992–7020, 2021.

[17] A. Irshad, H. F. Ahmad, B. A. Alzahrani, M. Sher, and S. A. Chaudhry, “An efficient and anonymous Chaotic Map based authenticated key agreement for multi-server architecture,” KSII Transactions on Internet and Information Systems, vol. 10, no. 12, pp. 5572–5595, 2016.

[18] F. Wu, X. Li, L. Xu, A. K. Sangaih, and J. J. P. C. Rodrigues, “Authentication protocol for distributed cloud computing: an explanation of the security situations for Internet-of-Things-
enabled devices,” *IEEE Consumer Electronics Magazine*, vol. 7, no. 6, pp. 38–48, 2018.

[19] T.-Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, and C.-M. Chen, “An authenticated key exchange protocol for multi-server architecture in 5G networks,” *IEEE Access*, vol. 8, Article ID 28096, 2020.

[20] T.-T. Truong, M.-T. Tran, A.-D. Duong, and I. Echizen, “Provable identity based user authentication scheme on ECC in multi-server environment,” *Wireless Personal Communications*, vol. 95, no. 3, pp. 2785–2801, 2017.

[21] K.-H. Yeh, “A provably secure multi-server based authentication scheme,” *Wireless Personal Communications*, vol. 79, no. 3, pp. 1621–1634, 2014.

[22] Y. Zhao, S. Li, and L. Jiang, “Secure and efficient user authentication scheme based on password and smart card for multi-server environment,” *Security and Communication Networks*, vol. 2018, Article ID 9178941, 13 pages, 2018.

[23] M. Hassan, A. Sultan, A. A. Awan, S. Tahir, and I. Ihsan, “An enhanced and secure multiserver-based user authentication protocol,” in *Proceedings of the International Conference on Cyber Warfare and Security (ICCWS)*, pp. 1–6, Islamabad, Pakistan, October 2020.

[24] M. Eltoweissy, S. Olariu, and M. Younis, “Towards autonomous vehicular clouds,” *Ad Hoc Networks*, vol. 49, pp. 1–16, 2010.

[25] S. Bitam, A. Mellouk, and S. Zeadally, “VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks,” *IEEE Wireless Communications*, vol. 22, no. 1, pp. 96–102, 2015.

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

[27] S. K. Bhoi, S. K. Panda, S. R. Ray et al., “TSP-HVC: a novel task scheduling policy for heterogeneous vehicular cloud environment,” *International Journal of Information Technology*, vol. 11, no. 4, pp. 853–858, 2019.

[28] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, “A secure authentication protocol for Internet of vehicles,” *IEEE Access*, vol. 7, Article ID 12047, 2019.

[29] J. Zhang, H. Zhong, J. Cui, and Y. Xu, “SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1810–1824, 2020.

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

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

[32] C. M. Chen, Y. Huang, K. H. Wang, and S. Kumari, “A secure authenticated and key exchange scheme for fog computing,” *Enterprise Information Systems*, 2020.

[33] M. Raya and J.-P. Hubaux, “Securing vehicular ad hoc networks,” *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

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

[35] M. C. Ming-Chin Chuang and J. F. Jeng-Farn Lee, “TEAM: trust-extended authentication mechanism for vehicular ad hoc networks,” *IEEE systems journal*, vol. 8, no. 3, pp. 749–758, 2014.
[51] R. Canetti, O. Goldreich, and S. Halevi, “The random oracle methodology, revisited,” *Journal of the ACM*, vol. 51, no. 4, pp. 557–594, 2004.

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

[53] B. Blanchet, “An efficient cryptographic protocol verifier based on prolog rules,” in *Proceedings 14th IEEE Computer Security Foundations Workshop*, pp. 82–96, Cape Breton, NS, Canada, June 2001.

[54] Y. Liu, Y. Wang, and G. Chang, “Efficient privacy-preserving dual authentication and key agreement scheme for secure V2V communications in an IoV paradigm,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2740–2749, 2017.