SMAKA: Secure Many-to-Many Authentication and Key Agreement Scheme for Vehicular Networks

Jing Zhang, Hong Zhong, Jie Cui, Yan Xu, Lu Liu

***Abstract*—With the rising popularity of the Internet and communication technology, vehicles can analyze and judge the real-time data collected by various cloud service providers (CSPs) in a vehicular network. However, in a vehicular network environment, real-time data are transmitted via wireless channels, which can lead to security and privacy issues. To avoid illegal access by adversaries, vehicle authentication and key agreement mechanism has been considered as one of the promising security measures in vehicular network environments. Besides, most of the solutions focus on authentication between one vehicle and one CSP. In such strategies, the implementation of efficient authentication for multiple vehicles and CSPs simultaneously is usually challenging. Further, they are also subjected to performance limitations due to the overhead incurred. To solve these issues, we propose a many-to-many authentication and key agreement scheme for secure authentication between multiple vehicles and CSPs. The proposed scheme can prevent unauthorized access and provide SK-security even if temporary information is leaked. To improve the service, the CSP only needs to broadcast an anonymous message periodically instead of having to generate a unique anonymous message for each of vehicles. Similarly, when a vehicle wants to request the services of m CSPs, it only needs to send one request message instead of m. Therefore, the proposed scheme not only implements many-to-many communication but also significantly reduces the computation and communication overhead. Moreover, a thorough security analysis shows that the proposed scheme provides better security compared to other related schemes.**

***Index Terms*—Vehicular Networks, Authentication, Security, Many-to-Many, Session Key (SK).**

I. Introduction

THE rapid growth in the number of vehicles and diverse user demand for services have led to an exponential growth in the data generated by vehicles [1], [2]. Therefore, cloud computing with the ability to collect, process, and share real-time data is widely used in vehicular networks [3]–[5]. To reduce data transmission delays and prevent single points of failure [6], efforts have been made to decentralize cloud services and instead work with multiple cloud service providers (CSPs). This has led to the emergence of multi-cloud environments in vehicular networks [7]. A two-layer vehicular network structure in a multi-cloud environment is shown in Figure 1. The lower layer consists of vehicles and base stations (BSs). The upper layer contains a registration authority (RA) and CSPs with different service functions. The entities can communicate with each other through wireless communication, such as IEEE 802.11p, 4G, and 5G. Although CSPs are expected to play an important role in vehicular network environments, there are some challenges in terms of data exchange with the vehicles.

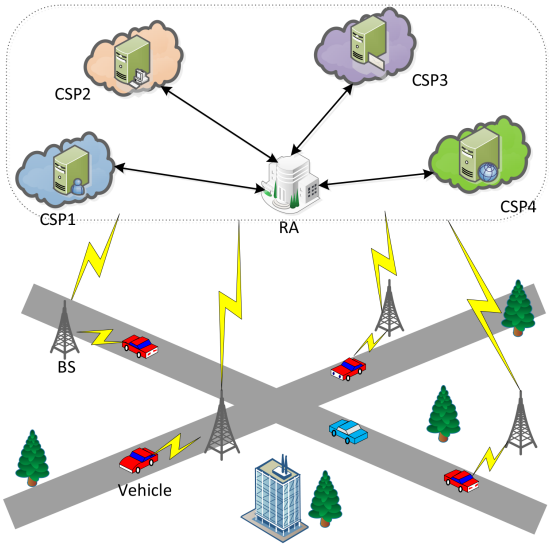


Fig. 1. Vehicular network structure in multi-cloud environments.

One of the challenges is the security and privacy issues that are introduced due to the use of wireless communication in vehicular network [8], [9]. For instance, sensitive and important transmission messages may be subjected to unauthorized access. Alternatively, if an adversary modifies, imitates, or replays the transmitted message, it may result in fatal harm to the data owner. The vehicular networks are accountable for accidents, which not only require user privacy protection but also the ability to trace the identities of the offender by authoritative institutions [10]. Therefore, to avoid illegal access to data and prevent malicious attackers, conditional privacy protection authentication and key agreement mechanisms are considered effective security measures.

Another challenge is the need for researchers to consider the large number of users apart from paying attention to diversified services. This challenge can be attributed to the parallel increase in the number of vehicles and service demands of the users [11], [12]. Currently, the existing solutions [7], [13], [14] focus not just on identity authentication of one vehicle to one server but also on the multiple service requirements

J. Zhang, H. Zhong, J. Cui, and Y. Xu are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, the School of Computer Science and Technology, Anhui University, Hefei 230039, China, the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China, and the Institute of Physical Science and Information Technology, Anhui University, Hefei 230039, China (e-mail: zhongh@ahu.edu.cn).

L. Liu is with the Department of Informatics, University of Leicester, LE1 7RH, UK (e-mail: l.liu@leicester.ac.uk).

of one vehicle to a particular server. However, these solutions are unaware that multiple vehicles require multiple services from multiple service providers simultaneously in actual deployment [15], [16]. Moreover, considering the large amount of overhead between different entities, it is difficult to apply the existing protocols to many-to-many scenarios.

A. Related Work

A series of research works have been done to design the authentication and key agreement protocol in vehicular networks, which can be divided into three categories. In 1981, Lamport [17] first introduced the concept of password authentication with insecure communication. Since then, several password authentication schemes have been proposed. The first category appears in various two-party authentication schemes [18]–[20]. However, these were only work in a single-server environment. Here, each user would need to be registered with each server separately. However, with the increase in the number of vehicles, the transmission delays and single point of failure of such solutions have become prominent. Therefore, it is impractical to directly apply the two-party authentication mechanism to the vehicular network environment.

After the conception of autonomous vehicular cloud in 2010 [21], cloud computing has gradually been applied to vehicular network environments. The second category is the emergence of several authentication schemes for vehicular cloud computing [3], [5], [22], [23]. Here, the cloud is of a temporary kind, which is composed of several voluntary vehicles. Although such schemes improve the utilization of vehicle's internal resources and reduce the service response time, they ignore the advantages of traditional cloud computing [24]. In particular, certain issues related to selection, management, task allocation, and the number of participating vehicles are not addressed.

To address these drawbacks, researchers have made efforts to extend a single traditional cloud to multiple cloud services [7], [25], [26] (or decentralizing the cloud to adopt fog computing [14], [27]–[29]). Here, these two types of schemes are collectively referred to as vehicles using cloud authentication schemes in a multi-cloud environment. In other words, the third category is a multi-cloud authentication scheme. Ma et al. [14] extended the solution proposed in [27] to support an efficient three-party authenticated key agreement protocol for fog-based vehicular ad-hoc networks. However, Chen et al. [28] pointed out that Jia et al.'s scheme [27] was vulnerable to an ephemeral secret leakage attack and proposed an enhanced scheme to withstand this attack. Further, we identified that Ma et al.'s scheme [14] also suffered from the same drawback as that of Jia et al.'s scheme, therefore, it also failed to prevent the ephemeral secret leakage attack.

Recently, Cui et al. [7] proposed an extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment to cope with the problem of CSP selection and resist known attacks. However, it has been noticed that the session key is generated by encrypting the anonymity of the vehicle, real identity of the CSP, and temporary information through a hash function. As the vehicle anonymity and hash function are public, the real identity of the CSP is revealed to the authenticated vehicle. Thus, the scheme [7] cannot withstand an ephemeral secret leakage attack. As a result, Ma et al.'s scheme [14] and Cui et al.'s scheme [7] cannot prevent impersonation and man-in-the-middle attacks. A disadvantage of the above-mentioned schemes is their lack of support toward many-to-many authentication scenarios. Therefore, it is necessary to propose a many-to-many authentication scheme to meet the security requirements and provide high computational efficiency.

B. Our Contribution

To address the above-mentioned challenges, a secure and efficient many-to-many authentication and key agreement mechanism, referred to as SMAKA, is proposed. Owing to the use of a hybrid encryption method and broadcast mechanism in the SMAKA scheme, a secure and efficient many-to-many communication is achieved. The contributions of our proposed scheme are as follows:

1) We investigate the implementation of data security and user privacy by establishing session keys in multi-party mutual authentication, and propose a secure authentication and key negotiation mechanism. Apart from restricting illegal access under various attacks, the proposed scheme also provides registration services in cases where a user accidentally leaks a private key. Session key security (SK-security) is supported even if temporary information is leaked.

2) We design a practical many-to-many authentication and key agreement scheme with an effective authentication function for simultaneous use in multiple vehicles and CSPs. In our proposed scheme, a vehicle selects multiple CSPs by sending a request message. In short, n vehicles requesting m CSPs would need to send only n request messages instead of nm. In addition, we use the broadcast mechanism on the CSP side. Each CSP needs to generate an anonymous/encrypted message that is broadcasted regularly, instead of having to generate an anonymous/encrypted message for each vehicle. To the best of our knowledge, this is the first solution that has been proposed with the above-mentioned features.

3) Security analysis proves that the proposed scheme can achieve session key security. The performance analysis results show that our scheme demonstrates satisfactory security and efficiency compared to other related schemes.

C. Organization of The Rest Paper

The structure of the rest paper is as follows: Section II introduces some preliminaries and system models. In Section III, we describe the proposed secure many-to-many authentication and key agreement scheme. The security analysis and performance evaluation of our scheme are outlined in section IV and V, respectively. Section VI gives the conclusion.

II. System Models and Objectives

In this section, we first describe the system and threat models to be used in the proposed SMAKA scheme for vehicular networks in multi-cloud environments. We then present some basic knowledge necessary for the proposed scheme.

A. Network Model and Assumptions

The many-to-many authentication and key agreement system based on multi-cloud service providers are shown in Figure 2. The system architecture comprises a registration authority (RA), multiple cloud service providers (CSPs), base stations (BSs), and numerous vehicles. The details of each component are described as follows:

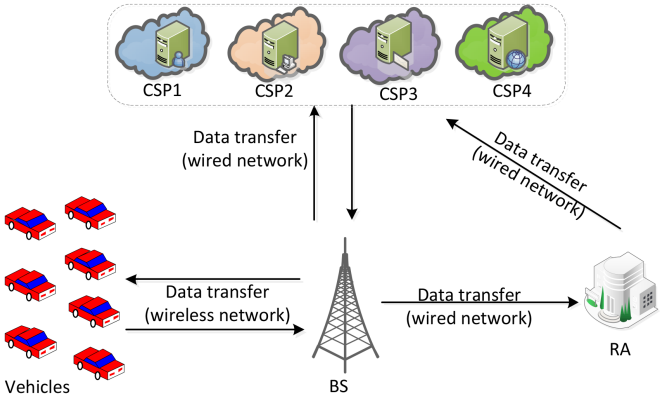


Fig. 2. System model of vehicular networks.

1) Vehicle: Each vehicle is equipped with an on-board unit (OBU) and a trusted platform module (TPM) [13]. The OBU is used for wireless communication with other vehicles or a BS, whereas the TPM is used to store security materials and handle cryptographic operations. Moreover, vehicles are assumed to have limited computing power.

2) BS: A BS is a wireless communication device deployed on the side of the road and is considered to be untrustworthy. It does not participate in any storage and computation and serves only as an intermediate transmission medium [7], [30]. The BS has a super-fast transmission speed to support seamless coverage for vehicle communication.

3) CSP: A CSP is an honest but curious entity that connects to the Internet in a wired manner and provides various network access services for vehicles. A TPM, which is responsible for storing security parameters and performing encryption/decryption operations, is integrated into the cloud computing system [31]. Different CSPs may provide different services. To improve traffic safety and convenience, the CSP periodically broadcasts safety- and entertainment-related services to nearby vehicles.

4) RA[1]***:*** The role of an RA, which is a highly secure entity that is fully trusted and uncompromisable, is generally undertaken by an intelligent transport system department of the government [29]. The RA is assumed to have sufficient computing and storage capabilities and is in charge of generating and publishing system parameters. During system registration, the RA cooperates with each vehicle and CSP and generates unique long-term private keys for them. To prevent the RA from being a single point of failure, a set of reliable servers and redundant RAs with identical functionalities and databases are installed [8], [32]. The RA is notably the only entity that can track the true identity of any vehicle.

Note that in our scheme, every city in a country is under an RA. For a given country (e.g., China) or political union of territories (e.g., European Union), a single root RA is established. Each RA is connected to the root and neighbor RAs via wired links [33], [34]. Within the area covered by the root RA, when a vehicle moves from one city to another, the credentials of the vehicle will be verified by the RA of the city at which the vehicle is originally registered, at the initiative of the RA of the city wherein the vehicle is currently roaming [35], [36]. For simplicity, we illustrated a single RA in Figure 2. In addition, when vehicles move between areas covered by different root RAs, one of two cases will apply: 1) Along borders of countries with strong diplomatic ties, mutual trust, and equivalent standards, cross-certification can be employed. 2) Along borders where these conditions do not apply, an incoming vehicle is registered with the root RA of the destination via a temporary vehicle registration process.

[1]Here, RA is composed of redundant RAs and a set of reliable servers, such as registration servers, key generation servers, authentication servers, and tracing servers.

B. Threat Model

In the proposed scheme, we use the famous Dolev–Yao threat model [37]. According to this model, an adversary $\mathcal{A}$ is able to read, modify, delete, forge, replay, or even insert false information through insecure public channels between two communication parties. In addition, according to the CK-adversary model [38], [39], $\mathcal{A}$ can not only perform all functions mentioned in the DY model, but also disclose the secret credentials, session state, and session keys during a session. Therefore, a vehicle authentication scheme designed for a vehicular network environment should ensure that even if secret credentials (such as session temporary secrets and session keys) are disclosed to $\mathcal{A}$, it should have minimal impact on the confidentiality of other credentials of the participating communication entity [40]. Therefore, we assume that in a vehicular network, the vehicle and CSP are untrusted participants, whereas the RA is trusted.

C. Preliminaries

Elliptic Curve Cryptosystem: Let E_q: $by^2 = x^3 + ax^2 + x(\mathrm{mod} q)$ be a non-singular elliptic curve over the finite field F_q, where $q > 3$ is a large prime, $a, b \in F_q$, and $b(a^2 - 4)(\mathrm{mod}\, q) \neq 0$. Let $\mathbb{G}$ be a cyclic group on E_q of prime order p.

Definition 1. Discrete Logarithm (DL) Problem: Given two random points $P, Q \in \mathbb{G}$, where $Q = xP$, $x \in \mathbb{Z}_p^*$, and $\mathbb{Z}_p^* = \{1, 2, \cdots, p-1\}$, it is difficult to calculate x from Q in a probabilistic polynomial time (PPT).

Definition 2. Computational Diffie–Hellman (CDH) Problem: Given points $P, xP, yP \in \mathbb{G}$, where $x, y \in \mathbb{Z}_p^*$, the advantage of any PPT adversary to calculate $xyP \in \mathbb{G}$ without the knowledge of x and y is negligible.

Definition 3. One-Way Collision-Resistant Hash Function: A one-way collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^l$ is a deterministic algorithm that takes as input an arbitrary length binary string and returns as output a binary string of length l [41], [42]. If $Adv_{\mathcal{A}}^{H}(t)$ is the advantage of

TABLE I
THE NOTATIONS AND DEFINITIONS USED

Notations	Definitions
RA	Trusted registration authority
V_i, CSP_j	i^{th} vehicle and j^{th} CSP, respectively
ID, CID	Real identities of vehicle and CSP, respectively
s	Master secret key of RA
P_{pub}	Public key of RA
s_i, s_j	Long-term secrets of V_i and CSP_j, respectively
PW_i, UID_i	Password and identity of user U_i, respectively
sk_i	Long-term session key between U_i and V_i
$k, \alpha, \beta, \gamma, \mu$	Random numbers
PID_i, PID_j	Pseudo-identities of V_i and CSP_j, respectively
TS	Current timestamp
ΔT	Validity period of message
$SK_{i,j}$	Session key between V_i and CSP_j
$E_k(\cdot)/D_k(\cdot)$	Symmetric encryption/decryption using key k
H, H_1	Two secure one-way hash functions
$\oplus, \|\|$	Exclusive-OR and concatenation operation
$UPW, \phi_i, A_i, B_i, C_i$	Authentication tokens for verifying UID, PW, s_i, and sk_i
$R_j, R_j^*, W_i, W_i^*, J_i$	Points on group $\mathbb{G}$
$\tau, S_j, F_i, K_r, N_r, W_i^{"}, GID_r, \theta, \rho, L_j$	Hashvalues
M	Message sent by V_i, CSP_j or RA

an adversary $\mathcal{A}$ at finding a hash collision in execution time t, $Adv_{\mathcal{A}}^{H}(t) = Pr[(x_1, x_2) \in_R \mathcal{A} : x_1 \neq x_2, H(x_1) = H(x_2)]$, where $(x_1, x_2) \in_R \mathcal{A}$ denotes that x_1 and x_2 are randomly selected by $\mathcal{A}$. An (ε, t)-adversary $\mathcal{A}$ attacking the collision resistance of $H(\cdot)$ indicates $Adv_{\mathcal{A}}^{H}(t) \leq \varepsilon$ with at most execution time t. Note that Adv refers to the advantage of breaking a scheme or solving a difficult problem.

III. PROPOSED SMAKA SCHEME

In this section, we describe the proposed SMAKA scheme for vehicular networks. Table I lists the main notations and corresponding definitions used in the five phases of this study. The first phase is system initialization, in which the RA allocates public parameters to the system. The second phase helps users and CSPs legally register with the RA to obtain the necessary credentials stored in their TPM. The third phase is a necessary preparation phase, in which the CSP generates a regular broadcast message and the vehicle completes a successful login. CSP broadcast is particularly necessary; it simplifies the selection of one CSP for multiple vehicles simultaneously and makes fully prepares for the realization of many-to-many selection services. The fourth phase helps V_i and CSP_j authenticate with each other and establishes a session key for secure communication. The entire authentication process not only allows a vehicle to select multiple CSP services through the sending of a message, but also prevents the real identity of each vehicle from being revealed to the CSP. In the last phase, we discuss how to update the identity or password of a vehicle. The security of our scheme is based on a Diffie–Hellman problem defined over a cyclic group. The Diffie–Hellman problem is widely believed to be secure in traditional Turing machine computational models only. Security against quantum computers is outside the scope of this study.

A. System Initialization

The RA processes the following steps.

1) The RA chooses a random number s as its master key and calculates its corresponding public key $P_{pub} = sP$, where P is a generator point of the group $\mathbb{G}$ with order p.

2) The RA chooses two one-way collision-resistant hash functions $H : \{0,1\}^* \rightarrow \{0,1\}^l$, $H_1 : \{0,1\}^* \rightarrow \{0,1\}^l$, where l denotes the output length of the hash function. It then publishes $params = \{\mathbb{G}, p, P, P_{pub}, H, H_1\}$ as the system parameters.

B. Registration Phase

Through the execution of this phase, vehicles and CSPs are registered with the RA in offline mode via a reliable channel (e.g., in-person).

- **Cloud Service Provider Registration**

Figure 3 shows the interactions between the CSP and RA during this registration phase.

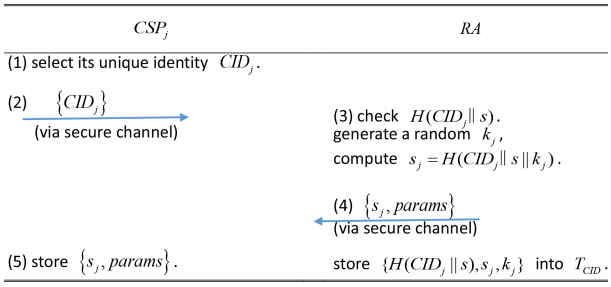


Fig. 3. Cloud service provider registration phase.

In this phase, CSP_j first chooses its unique identity CID_j and submits a registration request $\{CID_j\}$ to the registered RA through a secure channel. Upon receiving the request, the RA checks whether the hash value $H(CID_j||s)$ already exists in its identity-verifier database T_{CID}. If it already exists the request is rejected. Otherwise, the RA generates a random number k_j and computes $s_j = H(CID_j||s||k_j)$. The RA then delivers the tuple $\{s_j, params\}$ to each CSP. In

addition, the RA stores the tuple $\{\tau_j = H(CID_j||s), s_j, k_j\}$ into its identity-verifier database T_{CID}. After receiving the tuple, the CSP_j keeps s_j and $params$ secretly to complete the registration process. Thereafter, CSP_j is loaded with $\{s_j, params\}$.

- **Vehicle Registration**

Figure 4 shows the interactions between the vehicle user and the RA during the vehicle registration phase.

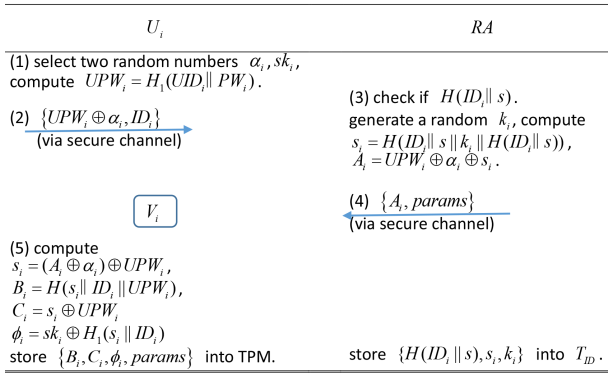


Fig. 4. Vehicle registration phase.

1) The user U_i is free to generate their identity UID_i, password PW_i, and computes $UPW_i = H_1(UID_i||PW_i)$. U_i then generates two random numbers α and sk_i, where sk_i is the long-term session key used for the user identity or password update (see Section III.E). U_i also submits the registration request $\{UPW_i \oplus \alpha, ID_i\}$ secretly to the RA.

2) Upon receiving the request message, the RA checks the availability of the hash value $H(ID_i\|s)$ in its identity-verifier database T_{ID}. If it is not available, the RA generates a random number k_i and computes $s_i = H(ID_i||s||k_i||H(ID_i\|s))$, $A_i = UPW_i \oplus \alpha_i \oplus s_i$. The RA then delivers the tuple $\{A_i, params\}$ into each vehicle's TPM. In addition, the RA stores the tuple $\{\tau_i = H(ID_i||s), s_i, k_i\}$ into its T_{ID}.

3) After receiving the tuple, V_i computes its secret $s_i = (A_i \oplus \alpha_i) \oplus UPW_i$, $B_i = H(s_i\|ID_i||UPW_i)$, $C_i = s_i \oplus UPW_i$, and $\phi_i = sk_i \oplus H_1(s_i||ID_i)$. V_i then replaces A_i with B_i and stores B_i, C_i, ϕ_i and $params$ into its OBU to complete the registration process. Thereafter, V_i is loaded with $\{B_i, C_i, \phi_i, params\}$.

Remark 1: To avoid a privileged-insider attack, the random secret α can be used in a request message $\{UPW_i \oplus \alpha, ID_i\}$. Even though a privileged-insider user of the RA as an insider attacker realizes the request $UPW_i \oplus \alpha$, without knowing the secret α, it will be difficult for them to obtain the encrypted real identity and password UPW_i. Therefore, the adversary will not know the secrets UID_i and PW_i.

Remark 2: The proposed scheme supports re-registration when a secret key is revealed. The purpose of using the random secrets k_i/k_j is to prevent legitimate users/CSPs from accidentally leaking their secret keys s_i/s_j, with which they cannot be re-registered. Legitimate users/CSPs only have to request to revoke their account and re-register them to the RA with the same identity ID_i/CID_j. For example, for any user, the RA only needs to reselect a random number k_i^{new} and recalculate a new private key $s_i^{new} = H(ID_i||s||k_i^{new}||H(ID_i\|s))$.

C. Login Phase

In this phase, the CSP_j first periodically broadcasts messages for vehicles in need of services. When a broadcast message is received from the CSP_j, and the user U_i acknowledges wanting to receive the service from the CSP_j, the process begins with the first step, which is to log in to the vehicle. The login process to vehicle V_i from the CSP_j is shown in Figure 5.

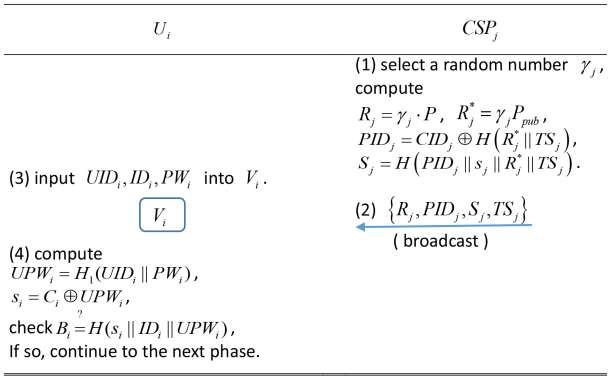


Fig. 5. User login phase.

1) The CSP_j selects a random number γ_j and generates an up-to-date timestamp TS_j. It then computes $R_j = \gamma_j \cdot P$, $R_j^* = \gamma_j \cdot P_{pub}$, a pseudo-identity $PID_j = CID_j \oplus H(R_j^*||TS_j)$, and $S_j = H(PID_j||s_j||R_j^*||TS_j)$. Finally, the CSP_j periodically broadcasts the message $M_j = \{R_j, PID_j, S_j, TS_j\}$ during the expiration of the timestamp TS_j. Note that the vehicle, RA, and CSP share a common trusted source for the timestamp because all entities are all installed with GPS devices.

2) Upon receiving the message M_j, the vehicle user U_i acknowledges that they want to receive the service from CSP_j. The vehicle user U_i inputs login credentials (such as UID_i, ID_i, and PW_i) into their V_i.

3) The vehicle V_i computes $UPW_i = H_1(UID_i||PW_i)$, $s_i = C_i \oplus UPW_i$, and checks whether the condition $B_i \stackrel{?}{=} H(s_i||ID_i||UPW_i)$ is true. If it is true, the TPM confirms that the user U_i is legitimate, and V_i continues to the next phase. Otherwise, V_i immediately rejects user U_i.

Remark 3: A CSP_j, which is responsible for broadcasting anonymous messages, may face simultaneous requests from multiple vehicles, so the authentication mechanism should be sufficiently efficient. In traditional methods, $N \cdot T_C$ is required to generate a pseudonym message for each vehicle, where N is the number of vehicles and T_C is the time overhead of CSP_j for generating an anonymous message. In this study, the broadcast method requires only T_C. This method also saves the authentication overhead of the RA. Similarly, the RA needs to verify CSP_j only once during the validity period of the message.

D. Authentication and Key Agreement Phases

Successful verification indicates that the user U_i has successfully logged in and completed the selection of a CSP from which they wish to access the services. V_i then performs an encryption calculation and sends the encrypted request

message M_1 to the RA instead of directly transmitting a unique request message to each CSP_j (such as [9], [14]). The RA checks the legitimacy of the vehicle V_i and helps V_i to verify the legitimacy of CSP_j. If both V_i and CSP_j are legitimate, the RA helps the vehicle V_i to send the request message M_2 to the corresponding CSP_j. Afterward, CSP_j and V_i complete authentication with the help of the RA and establish a session key for secure communication. Both the authentication and key agreement phases are presented in Figure 6.

1) The vehicle V_i first checks the timestamp TS_j of the message based on condition $|TS_j^* - TS_j| < \Delta T$, where the timestamp TS_j^* is the time that the V_i received the message M_j. If the condition is satisfied, the vehicle generates a random number β_i and the current timestamp TS_i. To prevent a temporary secret (β_i) leak attack, V_i computes $\mu_i = H(\beta_i||s_i||TS_i)$, $W_i = \mu_i \cdot P$, and $W_i^* = \mu_i P_{pub}$. To ensure security and save computing overhead, V_i calculates a pseudo-identity $PID_i = E_{W_{i,x}^*}(ID_i, PID_j, TS_j, TS_i)$ and a hash signature $F_i = H(PID_i||s_i||W_i^*||TS_i)$, where $W_{i,x}^*$ denotes the x-coordinate of the elliptic curve point W_i^*. Note that an encrypted pseudo-identity PID_i can encrypt the anonymous identities of multiple CSPs at a time. Finally, the vehicle V_i sends a request authentication message $M_1 = \{W_i, PID_i, F_i\}$ to the RA over an open channel.
2) When the message M_1 is received, the RA first computes $W_i^* = sW_i (= \mu_i P_{pub})$ and uses $W_{i,x}^*$ to decrypt PID_i to obtain ID_i, PID_j, TS_j, and TS_i. The RA checks the freshness of TS_i and validates ID_i. If V_i's real identity ID_i is not in the revocation list, the RA verifies the condition $F_i \stackrel{?}{=} H(PID_i||s_i||W_i^*||TS_i)$ by computing $s_i = H(ID_i||s||k_i||\tau_i)$. If it holds, the RA checks whether the corresponding anonymity PID_j, ID_j, TS_j already exists. If it already exists, the CSP_j has been verified. If it does not exist, the RA computes $R_j^* = sR_j (= \gamma_j P_{pub})$ to obtain CSP_j's real identity $CID_j = PID_j \oplus H(R_j^*||TS_j)$ and checks the condition $S_j \stackrel{?}{=} H(PID_j||s_j||R_j^*||TS_j)$. Note that the RA only needs to perform verification once within the corresponding anonymity PID_j validity period.
3) If this authentication fails, the RA rejects the legitimacy of V_i by denying the request message M_1. Otherwise, the RA confirms that the received credentials (ID_i, PID_j, TS_j) are valid, and generates a hash secret $K_r = H(ID_i||CID_j||TS_r)$ for the authentication of V_i and CSP_j. The RA then calculates the authentication message for CSP_j: $N_r = K_r \oplus H(R_j^*||s_j||TS_r||TS_j)$, $W_i^{"} = W_i \oplus H(K_r||R_j^*||TS_r||TS_j)$ and $\theta = H(K_r||GID_r||CID_j||s_j||R_j^*||TS_r)$. Simultaneously, the RA calculates the authentication message for V_i: $GID_r = CID_j \oplus H(ID_i||W_i^*||TS_r)$ and $\rho = H(K_r||GID_r||CID_j||s_i||W_i^*||TS_r)$. Finally, the RA transmits the message $M_2 = \{N_r, W_i^{"}, GID_r, TS_r, \theta, \rho\}$ to CSP_j.
4) Upon receiving the message M_2, CSP_j first checks the freshness of the timestamp TS_r. The receiving time is assumed to be $TS_r{}^*$. If $|TS_r^* - TS_r| < \Delta T$, CSP_j acquires $K_r = N_r \oplus H(R_j^*||s_j||TS_r||TS_j)$, and then checks whether the condition $\theta = H(K_r||GID_r||CID_j||H(CID_j||s)||R_j^*||TS_r)$ is true. If it is not true, CSP_j immediately stops the session. Otherwise, CSP_j computes the session key $SK_{ij} = H(J_i||K_r||CID_j||TS_{ij})$ and a hash signature $L_j = H(SK_{ij}||GID_r||\rho||CID_j||TS_{ij})$ by decrypting $W_i = W_i^{"} \oplus H(K_r||R_j^*||TS_r||TS_j)$ and computing $J_i = \gamma_j W_i (= \mu_i R_j)$. Finally, CSP_j transmits the message $M_3 = \{L_j, GID_r, \rho, TS_r, TS_{ij}\}$ via open channel.
5) V_i receives the message M_3 and checks the freshness of the timestamp TS_{ij}. If the verification holds, V_i computes $K_r = H(ID_i||CID_j||TS_r)$ by decrypting $CID_j = GID_i \oplus H(ID_i||W_i^*||TS_r)$ shared with the RA to verify the condition $\rho \stackrel{?}{=} H(K_r||GID_r||CID_j||s_i||W_i^*||TS_r)$. If the condition is true, V_i continues to compute the session key $SK_{ij}^* = H(J_i||K_r||CID_j||TS_{ij})$ by computing $J_i = \mu_i R_j (= \gamma_j W_i)$ shared with CSP_j to check the condition $L_j \stackrel{?}{=} H(SK_{ij}^*||GID_r||\rho||CID_j||TS_{ij})$. If the verification holds, V_i authenticates CSP_j.

Finally, both the vehicle V_i and the CSP CSP_j securely store the common session key $SK_{ij}^* = SK_{ij}$ for their future communications.

Remark 4: The authentication and key agreement process requires only three rounds because the message $M_2 = \{N_r, W_i^{"}, GID_r, TS_r, \theta, \rho\}$ from the RA contains factors s_j and CID_j, which cannot be tampered. Similarly, the message $M_3 = \{L_j, GID_r, \rho, TS_r, TS_{ij}\}$ from CSP_j contains the authentication information GID_r and ρ from the RA. GID_r and ρ contain the tamperable factors s_i, ID_i, and CID_j, which can help verify the message M_3 from CSP_j. Therefore, the proposed scheme can not only ensure message security, but also realize low-round authentication.

E. User Identity or Password Update Phase

If a vehicle is accessed by a new user (under the assumption that the new user is already registered), or if a user wants to update the password, then the following steps that are not involved with the RA should be executed.

1) U_i inputs their current identity UID_i, password PW_i, and vehicle identity ID_i into the vehicle's TPM. The vehicle V_i computes $UPW_i = H_1(UID_i||PW_i)$, $s_i = C_i \oplus UPW_i$ and checks whether the equation $B_i = H(s_i||ID_i||UPW_i)$ satisfied. Upon unsuccessful verification, this process is terminated by V_i. Otherwise, U_i is considered as the actual user.

2) After receiving instructions from the TPM, U_i selects a new password PW_i^{new}, and replaces the new identity UID_i^{new} of the new user if necessary. U_i then computes $UPW_i = H_1(UID_i||PW_i)$, $UPW_i^{new} = H_1(UID_i^{new}||PW_i^{new})$, and $UPW_i^* = H_1(UPW_i||UPW_i^{new}||sk_i)$. After that, U_i inputs the updated information $\{UPW_i^{new}, UPW_i^*\}$ into the vehicle's TPM. V_i computes $sk_i = \phi_i \oplus H_1(s_i||ID_i)$ and checks if the condition $UPW_i^* \stackrel{?}{=} H_1(UPW_i||UPW_i^{new}||sk_i)$ is true.

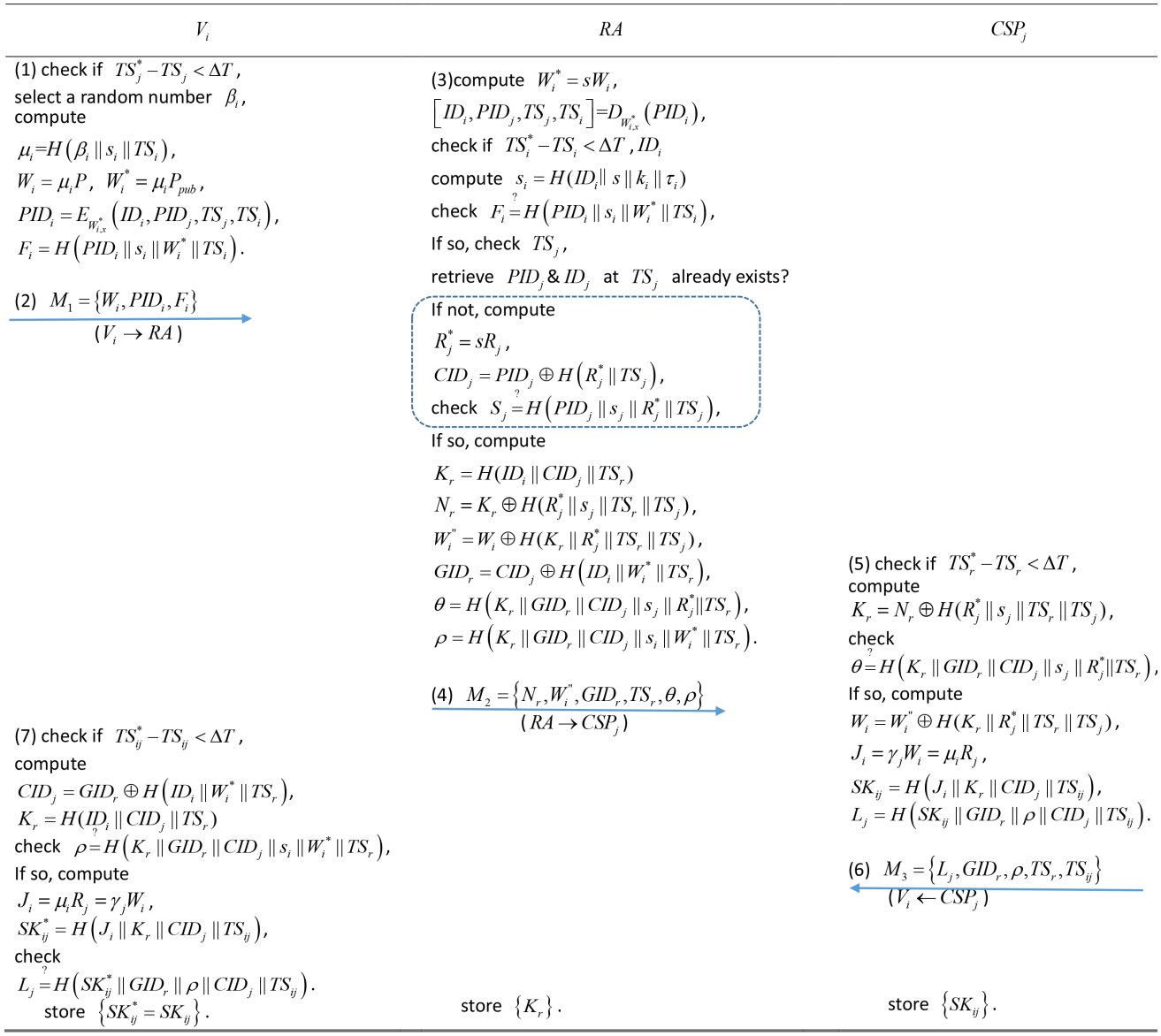


Fig. 6. Authentication and key agreement phases.

If true, V_i calculates $B_i^{new} = H(s_i||ID_i||UPW_i^{new})$ and $C_i^{new} = s_i \oplus UPW_i^{new}$.

3) At the end of this phase, the vehicle V_i replaces B_i with B_i^{new} and changes C_i to C_i^{new}.

Remark 5: The proposed scheme can be extended such that the user identity cannot be changed, for example, a person to a vehicle. Setting $s_i = H(ID_i||s||k_i||H(UID_i\|ID_i\|s))$ is necessary only at registration with the RA. Moreover, this extension can eliminate many-logged-in-user attacks. The table entry of user U_i can be expressed as $\{H(UID_i||s), k_i, status\}$, where $status \in \{-1, 0, 1\}$. Here, $status = 1$ indicates that the user is logged in and active; and $status = 0$ indicates that the user is not logged in but is active; and $status = -1$ denotes that the user is not logged in and is inactive. Therefore, the proposed scheme is highly malleable and can fulfil requirements for various situations.

IV. Security Proof and Analysis

This section mainly analyzes and proves the security of the proposed authentication scheme. First, we analyze the proposed scheme using the broadly-used real-or-random (ROR) model [7], [40], [43]–[45] and show that the proposed scheme provides secure authentication. Moreover, a security analysis on whether the proposed scheme can protect against other known attacks is presented.

A. Security Proof using ROR Model

Before we prove that session key security is preserved by the proposed scheme, the following primitives associated with the ROR model [43] are presented.

- **Participants.** Let V_i^a denote the instance a of a vehicle V_i. Analogously, ${CSP_j}^b$ denotes the instance b of the CSP CSP_j, and RA^c represents the instance c of the RA. These instances are referred to as oracles.

- **Accepted state.** If the instance V_i^a moves to the accepted state when the last expected protocol message is received, it will be in the accepted state. The session identification (sid) of V_i^a for the current session is constructed via concatenation of all communicated (received and sent) messages V_i^a in order.
- **Partnering.** If we say that instances V_i^a and ${CSP_j}^b$ are partners of each other, they need to meet the following conditions:1) both V_i^a and ${CSP_j}^b$ are in the accepted state; 2) V_i^a and ${CSP_j}^b$ authenticate each other mutually and share the same sid; and 3) both V_i^a and ${CSP_j}^b$ are mutual partners of each other.
- **Freshness.** If the session key SK_{ij} between the vehicle V_i and the CSP CSP_j is not revealed to the adversary $\mathcal{A}$, the instance V_i^a or ${CSP_j}^b$ is regarded as fresh.

The adversary $\mathcal{A}$ is assumed to have complete control over all communications in the vehicular network. Moreover, $\mathcal{A}$ can read (intercept) and modify all the exchanged messages and can forge new messages into the vehicular network. $\mathcal{A}$ is capable of the following oracle queries:

- $Update(V_i^a, UPW)$: This query models a man-in-the-middle attack in which updated information UPW is intercepted by the adversary $\mathcal{A}$.
- $Execute(V_i^a, {CSP_j}^b, RA^c)$: This query models an eavesdropping attack. The attacker $\mathcal{A}$ can read (intercept) all messages $\{M_j, M_1, M_2, M_3\}$ during communications among V_i, CSP_j, and RA.
- $Reveal(V_i^a, {CSP_j}^b)$: When this query is performed, the session key SK_{ij} established between V_i^a and its partner ${CSP_j}^b$ is revealed to the adversary $\mathcal{A}$.
- $Send(V_i^a, M)$: This query models an active attack. The adversary $\mathcal{A}$ can send a message M to a participant instance V_i^a and receive a response message.
- $Corrupt(V_i^a)$: By performing this query, the adversary $\mathcal{A}$ can obtain all the secret parameters stored in TPM_i of the registered participant instance V_i^a.
- $Test(V_i^a, {CSP_j}^b)$: Under this query, the semantic security of the session key $SK_{i,j}$ is simulated. At the beginning of the experiment, a coin c flips to a value that is closely related to the adversary $\mathcal{A}$ and plays a decisive role in the output of this query. If the session key has not been established or instance V_i^a (or ${CSP_j}^b$) is not fresh, a null value($\perp$) is returned. On the contrary, if $c = 1$, the instance V_i^a (or ${CSP_j}^b$) returns the session key $SK_{i,j}$ to the adversary $\mathcal{A}$, whereas if $c = 0$, it returns a random number to the adversary $\mathcal{A}$.

Note that all participants, including the adversary $\mathcal{A}$, can access hash functions $H(\cdot)$ and $H_1(\cdot)$ that H and H_1 are modeled by a random oracle, respectively.

B. Formal Security Proof

We use the sequences of games approach [46] for our formal security proof of the proposed authentication and key agreement (AKA) scheme. First, we list the following lemma given by Shoup [46]:

Lemma 1 (Difference Lemma). Let A, B, F denote events defined in some probability space, and assume that $A \wedge \neg F \iff B \wedge \neg F$. Thus $|\Pr[A] - \Pr[B]| \leq \Pr[F]$.

Theorem 1. Let $\mathcal{A}$ be the adversary running in polynomial time t against the proposed AKA scheme. D is a uniformly distributed password dictionary with a size of $|D|$. q_s, q_e, q_h, and q_{h_1} represent the numbers of $Send$ queries, $Execution$ queries, H queries, and H_1 queries, respectively. $|H|$, $|H_1|$, and l denote the range spaces of the H, H_1, and the length of hash values, respectively. $Adv_{\mathcal{A}}^{CDH}(t)$ is the advantage of the adversary $\mathcal{A}$ at breaking the CDH problem in the upper-bound time t. The advantage of $\mathcal{A}$ at breaking the session key security of the proposed AKA scheme can then be estimated as

$$Adv_{\mathcal{A}}^{AKA}(t) \leq \frac{q_h^2 + q_{h_1}^2 + (q_s + q_e)^2}{2p} + \frac{q_s^2}{2^l} + \frac{q_s}{(2^l |D|)} + \frac{q_{h_1}^2}{(2 |H_1|)} + \frac{q_h}{p} + q_s Adv_{\mathcal{A}}^{CDH}(t) \tag{1}$$

Proof: We construct a challenger $\mathcal{C}$ who can solve the CDH problem with a non-negligible probability by running $\mathcal{A}$ as a subroutine. Given an instance (P, $X = xP$, $Y = yP$) of the CDH problem, $\mathcal{C}$'s goal is to compute $Z = xyP$. $\mathcal{C}$ randomly picks $s \in \mathbb{Z}_p^*$, computes $P_{pub} = sP$, and publishes the system parameters $params = \{\mathbb{G}, p, P, P_{pub}, H, H_1\}$ to $\mathcal{A}$. $\mathcal{C}$ sets the identity, password and the TPM's secret parameters for each vehicle user. The proof consists of a sequence of games G_0-G_4. SC_{G_i} is used to indicate that the adversary $\mathcal{A}$ can succeed in guessing the value of c in a game G_i.

$Game\ G_0$: This game G_0 simulates the original attack. In this game, $\mathcal{C}$ simulates the oracle queries as a real player would. Therefore, in this experiment, the probability of success is equal to the probability that the adversary $\mathcal{A}$ successfully attacks the real protocol. According to the definition of semantic security [43], [44], we can obtain

$$Adv_{\mathcal{A}}^{AKA}(t) = |2 \cdot \Pr[SC_{G_0}] - 1| . \tag{2}$$

$Game\ G_1$: G_1 simulates all the oracles in G_0, except that $\mathcal{C}$ maintains the lists to store the answers of the oracles. Upon receiving $\mathcal{A}$'s query with the message M, $\mathcal{C}$ checks the corresponding list and sends the value to $\mathcal{A}$ if there is an entry already. Otherwise, $\mathcal{C}$ generates a random number, adds the value to the corresponding list, and sends the value to $\mathcal{A}$. The simulations of all the queries are listed in Table II. From the properties of random oracles, G_1 is observed to be indistinguishable from G_0. Thus, we have

$$\Pr[SC_{G_0}] = \Pr[SC_{G_1}] . \tag{3}$$

$Game\ G_2$: G_2 simulates all the oracles in G_1, except that $\mathcal{C}$ will terminate if the following two events occur. Based on the birthday paradox, it is clear that:

- Event E_1: The maximum probability is $\frac{q_h^2 + q_{h_1}^2}{2p}$ for a collision that occurs on the output of two hash functions H and H_1.
- Event E_2: The maximum probability is $\frac{(q_s+q_e)^2}{2p}$ for a collision occurring on the copy of the messages

TABLE II
THE SIMULATION OF ORACLES

For a hash oracle query, $\mathcal{C}$ returns the hash value to $\mathcal{A}$ if a record is found in thelist L_H. Otherwise, $\mathcal{C}$ answers a randomly chosen string from $\{0,1\}$, and adds the value to the list L_H. If this query is asked by $\mathcal{A}$, $\mathcal{C}$ adds the record into the list L_{answer}.
For a $Send(SCP_j^b, START)$ query, $\mathcal{C}$ randomly chooses $\gamma_j \in \mathbb{Z}_p^*$, and computes $R_j = \gamma_j P$, $R_j^* = \gamma_j \cdot P_{pub}$, $PID_j = CID_j \oplus H(R_j^* \| TS_j)$, $S_j = H(PID_j \| s_j \| R_j^* \| TS_j)$ and sends the message $M_j = \{R_j, PID_j, S_j, TS_j\}$ to $\mathcal{A}$.
For a $Send(V_i^a, M_j)$ query, $\mathcal{C}$ randomly chooses $\beta_i \in \mathbb{Z}_p^*$, and computes $\mu_i = H(\beta_i \| s_i \| TS_i)$, $W_i = \mu_i \cdot P$, $W_i^* = \mu_i P_{pub}$, $PID_i = E_{W_{i,x}^*}(ID_i, PID_j, TS_j, TS_i)$, $F_i = H(PID_i \| s_i \| W_i^* \| TS_i)$ and sends the message $M_1 = \{W_i, PID_i, F_i\}$ to $\mathcal{A}$.
For a $Send(RA^c, M_1)$ query, $\mathcal{C}$ verifies the correctness of S_j and F_i. If both of them are correct, $\mathcal{C}$ computes $K_r = H(ID_i \| CID_j \| TS_r)$, $N_r = K_r \oplus H(R_j^* \| s_j \| TS_r \| TS_j)$, $W_i^{"} = W_i \oplus H(K_r \| R_j^* \| TS_r \| TS_j)$, $\theta = H(K_r \| GID_r \| CID_j \| s_j \| R_j^* \| TS_r)$, $GID_r = CID_j \oplus H(ID_i \| W_i^* \| TS_r)$, $\rho = H(K_r \| GID_r \| CID_j \| s_i \| W_i^* \| TS_r)$ and sends the message $M_2 = \{N_r, W_i^{"}, GID_r, TS_r, \theta, \rho\}$ to $\mathcal{A}$.
For a $Send(SCP_j^b, M_2)$ query, $\mathcal{C}$ verifies the correctness of θ. If it holds, $\mathcal{C}$ computes the session key $SK_{ij} = H(J_i \| K_r \| CID_j \| TS_{ij})$ and a hash signature $L_j = H(SK_{ij} \| GID_r \| \rho \| CID_j \| TS_{ij})$. Then, $\mathcal{C}$ sends the message $M_3 = \{L_j, GID_r, \rho, TS_r, TS_{ij}\}$ to $\mathcal{A}$.
For a $Send(V_i^a, M_3)$ query, $\mathcal{C}$ checks the validity of L_j. If it is not valid, $\mathcal{C}$ terminates the session. Otherwise, SCP_j^b is authenticated and share the same session key SK_{ij}. $\mathcal{C}$ adds (M_j, M_1, M_2, M_3) into the list L_{answer}.
For a $Corrupt(V_i^a, TPM)$ query, $\mathcal{C}$ returns the TPM's secret parameter to $\mathcal{A}$.
For a $Update(V_i^a, UPW)$ query, $\mathcal{C}$ returns the updated information to $\mathcal{A}$.
For a $Execute(V_i^a, {CSP_j}^b, RA^c)$ query, $\mathcal{C}$ recovers (M_j, M_1, M_2, M_3) from the list L_{answer} and returns $\mathcal{A}$.
For a $Reveal(V_i^a, {CSP_j}^b)$ query, $\mathcal{C}$ sends the session key SK_{ij} if the session instance V_i^a is accepted, else returns $null$.
For a $Test(V_i^a, {CSP_j}^b)$ query, $\mathcal{C}$ sends SK_{ij} if $c = 1$, else returns a random number with the same size as SK_{ij}.

(M_j, M_1, M_2, M_3), because the nonce γ_j and β_i are uniformly randomized.

Based on Lemma 1, we then have

$$|\Pr[SC_{G_2}] - \Pr[SC_{G_1}]| \leq \frac{q_h^2 + q_{h_1}^2 + (q_s + q_e)^2}{2p}. \quad (4)$$

Game G_3: In this game, if the adversary $\mathcal{A}$ can fake $\langle S_j, F_i, \theta, \rho, L_j \rangle$ without making the random oracle queries, the scheme is simply terminated. This situation only appears in the $Send$ queries. As a result, Games G_3 and G_2 are perfectly indistinguishable unless the vehicle rejects S_j or L_j, or the RA rejects F_i, or the CSP rejects θ or ρ. Thus, we have

$$|\Pr[SC_{G_3}] - \Pr[SC_{G_2}]| \leq \frac{q_s^2}{2^l}. \quad (5)$$

Game G_4: This game modifies the $Send$ query. $\mathcal{C}$ randomly picks a matched instance $(V_i^a, {CSP_j}^b, RA^c)$ and answers $\mathcal{A}$'s $Send$ queries as follows:

1) Upon receiving $\mathcal{A}$'s $Send(SCP_j^b, START)$ query, $\mathcal{C}$ sets $R_j = X$, $R_j^* = sR_j$, and generates PID_j, S_j, TS_j as in game G_3. Finally, $\mathcal{C}$ sends the message $M_j = \{R_j, PID_j, S_j, TS_j\}$ to $\mathcal{A}$.
2) Upon receiving $\mathcal{A}$'s $Send(V_i^a, M_j)$ query, $\mathcal{C}$ sets $W_i = Y$, $W_i^* = sW_i$, and generates PID_i, F_i as in game G_3. $\mathcal{C}$ returns $M_1 = \{W_i, PID_i, F_i\}$ to $\mathcal{A}$.
3) Upon receiving $\mathcal{A}$'s $Send(RA^c, M_1)$ query, $\mathcal{C}$ verifies S_j, F_i, and generates K_r, N_r, $W_i^{"}$, GID_r, TS_r, θ, ρ as in game G_3. $\mathcal{C}$ sends the message $M_2 = \{N_r, W_i^{"}, GID_r, TS_r, \theta, \rho\}$ to $\mathcal{A}$.
4) Upon receiving $\mathcal{A}$'s $Send(SCP_j^b, M_2)$ query, $\mathcal{C}$ sets $J_i = \gamma_j Y$, $SK_{ij} = H(J_i \| K_r \| CID_j \| TS_{ij})$ and generates $L_j, GID_r, \rho, TS_r, TS_{ij}$ as in game G_3. $\mathcal{C}$ sends $M_3 = \{L_j, GID_r, \rho, TS_r, TS_{ij}\}$ to $\mathcal{A}$.
5) Upon receiving $\mathcal{A}$'s $Send(V_i^a, M_3)$ query, $\mathcal{C}$ sets $J_i = \mu_i X$, $SK_{ij}^* = H(J_i \| K_r \| CID_j \| TS_{ij})$ and aborts the instance.

Suppose a differentiator that can successfully distinguish G_4 and G_3 exists. $\mathcal{C}$ could use this differentiator to solve the CDH problem. According to the description, $\mathcal{C}$ simulates all the queries without knowing x, y. The differentiator interacts with game G_3 if $Z = xyP \in \mathbb{G}$ is true, and $\mathcal{C}$ outputs 1. Otherwise, the differentiator interacts with G_4, and $\mathcal{C}$ outputs 0. The differentiator selects an instance with a probability of $\frac{1}{q_s}$. Thus, we have

$$|\Pr[SC_{G_4}] - \Pr[SC_{G_3}]| \leq q_s Adv_{\mathcal{A}}^{CDH}(t). \quad (6)$$

In this game, $SK_{ij}^* = H(Z \| K_r \| CID_j \| TS_{ij})$ is a random value that is independent of the password and the number x, y. $\mathcal{A}$ may distinguish a true SK_{ij} and a random number if the following events occur:

- Event E_3: $\mathcal{A}$ successfully impersonates the user and forges a message $M_1 = \{W_i, PID_i, F_i\}$. To achieve this, $\mathcal{A}$ must correctly calculate the values of PID_i and F_i. $\mathcal{A}$ makes the $Corrupt(V_i^a)$ query to obtain all the secret parameters $\{B_i, C_i, \phi_i, params\}$. If the adversary $\mathcal{A}$ wants to guess ID_i and PW_i of the user U_i from $B_i = H(s_i \| ID_i \| H_1(UID_i \| PW_i))$ and $C_i = s_i \oplus H_1(UID_i \| PW_i)$, then $\mathcal{A}$ needs to know both the secret key s_i and the user identity UID_i. If $\mathcal{A}$ executes q_s times $Corrupt$ queries for guessing ID_i/PW_i or matching s_i and UID_i, the probability that $\mathcal{A}$ outputs a valid M_1 is
$$Pr[E_3] \leq \frac{q_s}{(2^l |D|)}. \quad (7)$$
- Event E_4: Similar to event E_3, $\mathcal{A}$ is allowed to ask the $Update(V_i^a, UPW)$ query. $\mathcal{A}$ can obtain updated information $\{UPW_i^{new}, UPW_i^*\}$ via a man-in-the-middle attack. However, $\mathcal{A}$ cannot obtain the new PW_i^{new} and UID_i^{new} because the updated information $\{UPW_i^{new}, UPW_i^*\}$ is constructed using sk_i, and no hash collision occurs when $\mathcal{A}$ makes the $Update(V_i^a, UPW)$ query with help of H_1 query (see Definition 3). Thus, the probability that $\mathcal{A}$ outputs a valid M_1 is
$$Pr[E_4] \leq \frac{q_{h_1}^2}{(2 |H_1|)}. \quad (8)$$
- Event E_5: $\mathcal{A}$ successfully impersonates the CSP and forges a message $M_3 = \{L_j, GID_r, \rho, TS_r, TS_{ij}\}$. To achieve this, $\mathcal{A}$ must correctly calculate the values of L_j. $\mathcal{A}$ makes the $Corrupt(CSP_j^b)$ query to obtain all the secret parameters $\{s_j, params\}$. Thus, the probability that $\mathcal{A}$ outputs a valid M_3 is
$$Pr[E_5] \leq \frac{q_h}{p}. \quad (9)$$

Thus, we can calculate

$$\Pr[SC_{G_4}] = \frac{1}{2} + \frac{q_s}{(2^l |D|)} + \frac{q_{h_1}^2}{(2 |H_1|)} + \frac{q_h}{p}. \quad (10)$$

Based Equations (2)-(6) and Equation (10), we obtain the result

$$Adv_{\mathcal{A}}^{AKA}(t) \leq \frac{q_h^2 + q_{h_1}^2 + (q_s + q_e)^2}{2p} + \frac{q_s^2}{2^l} + \frac{q_s}{(2^l\,|D|)} + \frac{q_{h_1}^2}{(2\,|H_1|)} + \frac{q_h}{p} + q_s Adv_{\mathcal{A}}^{CDH}(t) \tag{11}$$

C. Other Possible Attacks

In this subsection, we discuss non-mathematically that our scheme can withstand various known attacks.

1) *Privileged Insider Attack*: During the vehicle registration, a legal user U_i submits the registration request message $\{UPW_i \oplus \alpha, ID_i\}$ to the registered RA, where ID_i, UPW_i, and α are the identity of vehicle V_i, the pseudo-password $H_1(UID_i||PW_i)$, and a random number, respectively. Because of the randomness of α and the unidirectionality of $H_1(\cdot)$, it will be difficult for privileged insiders in RA to obtain UID_i and PW_i. Therefore, the proposed scheme could resist a privileged insider attack.

2) *Offline Password Guessing Attack*: In the vehicle registration phase, the password UPW_i of a user U_i is involved in $s_i = H(ID_i||s||k_i||H(ID_i\|s))$, $B_i = H(s_i\|ID_i||H(UID_i\|PW_i))$ and $C_i = s_i \oplus H_1(UID_i||PW_i)$, which are stored in the TPM of V_i. The adversary $\mathcal{A}$ can adopt power analysis attacks [47] to extract all information including B_i, C_i, and s_i. However, without knowledge of the user identity UID_i and the real identity ID_i of the vehicle V_i, guessing the encrypted password PW_i becomes a computationally infeasible problem for the adversary $\mathcal{A}$.

3) *Mutual Authentication*: In the authentication and key agreement phase, mutual authentication among U_i, RA and CSP_j is achieved through the following three ways: (1) RA checks $F_i \stackrel{?}{=} H(PID_i||s_i||W_i^*||TS_i)$ and $S_j \stackrel{?}{=} H(PID_j||s_j||R_j^*||TS_j)$ to authenticate V_i and CSP_j, respectively; (2) CSP_j verifies the condition $\theta \stackrel{?}{=} H(K_r||GID_r||CID_j||s_j||R_j^*||TS_r)$ to authenticate RA directly, and validates ID_i of V_i indirectly; and (3) V_i checks $\rho \stackrel{?}{=} H(K_r||GID_r||CID_j||s_i||W_i^*||TS_r)$ to verify RA indirectly, and verifies the condition $L_j \stackrel{?}{=} H(SK_{ij}^*||GID_r||\rho||CID_j||TS_{ij})$ to validate CSP_j directly to establish the session key.

4) *Anonymity*: The real identity ID_i of the vehicle V_i is hidden in the pseudo-identity $PID_i = E_{W_{i,x}^*}(ID_i, PID_j, TS_j, TS_i)$, where $\mu_i = H(\beta_i||s_i||TS_i)$, $W_i = \mu_i \cdot P$, and $W_i^* = \mu_i P_{pub}$. The prerequisite for the adversary $\mathcal{A}$ to obtain the real identity ID_i of the vehicle V_i is knowing the temporary secret information μ_i, or β_i, s_i, and TS_i. The adversary $\mathcal{A}$ cannot compute ID_i because of the difficulty of solving DL and CDH problems. Hence, we conclude that the proposed scheme provides anonymity.

5) *Traceability*: When the vehicle misbehaves, the RA can trace its real identity by analyzing its messages. The real identity ID_i of the vehicle V_i is involved in the pseudo-identity $PID_i = E_{W_{i,x}^*}(ID_i, PID_j, TS_j, TS_i)$ generated by the vehicle V_i, where $W_i = \mu_i \cdot P$, and $W_i^* = \mu_i P_{pub} (= sW_i)$. Only the RA can decrypt the real identity ID_i of the vehicle by computing $[ID_i, PID_j, TS_j, TS_i] = D_{W_{i,x}^*}(PID_i)$. Thus the conditional privacy [48] of our proposed scheme is guaranteed.

6) *Un-linkability*: Each pseudo-identity PID_i and message M_i of V_i and CSP_j are encrypted by random numbers and timestamps, and each pseudo-identity and message is different. Because of the randomness of the random number and the timestamp, the attacker cannot distinguish which two different messages come from the same vehicle.

7) *Perfect Forward Secrecy*: In the proposed scheme, the session key $SK_{ij} = H(J_i||K_r||CID_j||TS_{ij}) = H(H(\beta_i||s_i||TS_i)\gamma_j P||H(ID_i||CID_j||TS_r)||CID_j||TS_{ij})$ is computed using the long-term temporal secret s_i of V_i, the real identity ID_i and CID_j of the V_i and CSP_j, the random numbers β_i and γ_j selected by CSP_j and V_i, and the fresh timestamp TS_i. Therefore, based on an assumption that the secret keys of all participants have been compromised because of the difficulty of solving the CDH problem, calculating SK_{ij} is also computationally infeasible for the attacker $\mathcal{A}$ without knowing the attributes s_i, β_i, TS_i, γ_j, ID_i, and CID_j. Consequently, the proposed scheme accomplishes forward secrecy.

8) *Impersonation Attacks*: The proposed scheme is resistant to impersonation attacks in the following two cases:

Case 1. Vehicle impersonation attack: Suppose the valid message $M_1 = \{W_i, PID_i, F_i\}$ sent by V_i is captured by the attacker $\mathcal{A}$. The attacker $\mathcal{A}$ tries to obtain useful information from the captured message M_1. The purpose of the attack by $\mathcal{A}$ is to use this information to generate a legitimate request message M_1 to deceive the authentication of the RA. However, the ID_i, s_i, β_i, and TS_i contained in the message M_1 are not known by the attacker $\mathcal{A}$. Therefore, $\mathcal{A}$ cannot generate a valid request message M_1; that is, our proposed scheme can resist a vehicle impersonation attack.

Case 2. CSP impersonation attack: Suppose the valid message $M_3 = \{L_j, GID_r, \rho, TS_r, TS_{ij}\}$ sent by CSP_j is captured by the attacker $\mathcal{A}$. $\mathcal{A}$ tries to obtain useful information from the captured message M_3. The purpose of the attack is to use this information to generate a legal response message M_3. However, the CID_j, γ_j, J_i, K_r, and SK_{ij} contained in the message M_3 are not revealed to $\mathcal{A}$. Thus, $\mathcal{A}$ cannot calculate a legal message M_3 in polynomial time; in other words, our proposed scheme can resist CSP impersonation attacks.

9) *Replay Attack*: We assume that the attacker $\mathcal{A}$ can monitor communications among V_i, RA, and CSP_j. Although the adversary can intercept the message, the message contains a timestamp with a short usage period and a random number with strong randomness. Therefore, the proposed scheme is resilient against a replay attack.

10) *Man-in-the-middle Attack*: Assume that $\mathcal{A}$ can intercept the transmitted messages M_1, M_2, and M_3 during the authentication and key agreement phases. The attacker $\mathcal{A}$ attempts to modify the arbitrary messages to deceive V_i, RA and CSP_j. For this purpose, $\mathcal{A}$ needs to gains the secret attributes ID_i, s_i, β_i, and TS_i to generate a legitimate request message M_1. For the same reason, $\mathcal{A}$ also cannot modify the other messages M_2 and M_3. Consequently, our proposed scheme can resist a man-in-the-middle attack.

11) *Ephemeral Secret Leakage (ESL) Attack*: In our scheme, both the vehicle V_i and CSP_j build a common session key as $SK_{ij} = H(J_i||K_r||CID_j||TS_{ij}) = H(H(\beta_i||s_i||TS_i)\gamma_j P||H(ID_i||CID_j||TS_r)||CID_j||TS_{ij})$. As in the DY model (see Section II.B), we expect the following two cases:

Case 1. Suppose the ephemeral secrets β_i, γ_j, and TS_i are revealed to the attacker $\mathcal{A}$. However, the long-term secrets ID_i, s_i and CID_j contained in the common session key SK_{ij} are not known by the attacker $\mathcal{A}$. Thus, $\mathcal{A}$ cannot create a valid session key SK_{ij}.

Case 2. Assume that some or all the long-term secrets ID_i, s_i, and CID_j are known to $\mathcal{A}$. Generating a valid session key SK_{ij} will still be a significantly difficult task for $\mathcal{A}$ without the ephemeral secrets β_i, γ_j and TS_i.

As mentioned, if $\mathcal{A}$ wants to derive the session key SK_{ij}, $\mathcal{A}$ must possess both the ephemeral secrets and long-term secrets. Moreover, even if a particular session key SK_{ij} is compromised, the session keys SK_{ij} generated in previous or future sessions are completely different because of both the long-term secrets and the freshness of the temporary secrets. Thus, our scheme satisfies both forward and backward secrecy along with the session key security.

V. Performance Comparison

In this section, we discuss comparisons of computation and communication overheads in the authentication and key agreement phases of the proposed scheme and other existing related schemes, such as those by Liu et al. [13], Ma et al. [14], and Cui et al. [7]. Generally, the CSP and user registration phases are executed only once. Therefore, we focus only on the login, authentication and key agreement phases. In particular, we consider the complexities of one-to-one and many-to-many communication types in terms of communication and computation overhead.

For a bilinear pairing-based scheme [13], to achieve a security level of 128 bits, we construct a bilinear pairing $\overline{e}$: $G_1 \times G_1 \rightarrow G_T$, where G_1 is an additive group that is generated by a point $\overline{P}$ with order $\overline{p}$ on a supersingular elliptic curve $\overline{E}$: $y^2 = x^3 - 3x \bmod \overline{q}$ of embedding degree 2, where $\overline{q}$ is a 1536-bit prime number, and $\overline{p}$ is a 256-bit prime number. For Weierstrass curve-based schemes [7], [14], we construct an additive group G generated by a point $\tilde{P}$ with order $\tilde{p}$ on a non-singular elliptic curve $\tilde{E}$: $y^2 = x^3 + ax + b(\mathrm{mod}\tilde{q})$ to achieve a security level of 128 bits, where $\tilde{p}, \tilde{q}$ are two 256 bit prime numbers. For the Montgomery curve-based scheme (the proposed scheme), we construct an additive group $\mathbb{G}$ generated by a point P with order p on a non-singular elliptic curve E: $by^2 = x^3 + ax^2 + x(\bmod q)$ to achieve a security level of 128 bits, where p, q are two 256 bit prime numbers.

A. Comparison of Computation Overheads

To facilitate the comparison of computation overheads between the proposed scheme and other existing related schemes, we first introduce the experimental computation times required for various cryptographic operations. Let T_{bp}, $T_{bp.m}$, T_{mtp}, $T_{W.m}$, $T_{M.m}$, T_h, and $T_{e/d}$ respectively represent the time to execute a bilinear pair operation, bilinear pair multiplication operation, MapToPoint hash operation, Weierstrass curve scale multiplication operation, Montgomery curve scale multiplication operation, one-way hash function operation, and symmetric encryption/decryption (using AES-CBC).

We note that the computational overhead estimation might not be realistic and may be different for different platforms. In our experiments, we used a personal computer (HP with an Intel(R) Core(TM) i7-6700@ 3.4GHz processor, 8GB main memory, and the Ubuntu 14.04 operation system) as the server (CSP and RA), and used a mobile phone (Samsung Galaxy S5 with a Quad-core 2.45G processor, 2GB memory, and the Google Android 4.4.2 operating system) as the vehicle. Moreover, we have executed these operations 5000 times to obtain the average running time based on the MiRACL library [49]. The results are listed in Table III. Note that we neglect the execution time of the bitwise XOR operation because it is observably light compared with those of other operations.

TABLE III
Execution times of cryptographic operations (millisecond)

Operations	T_{bp}	$T_{bp.m}$	T_{mtp}	$T_{W.m}$	$T_{M.m}$	T_h	$T_{e/d}$
Vehicle time	32.713	9.405	0.732	2.211	1.341	0.056	0.162
Sever time	5.086	0.694	0.099	0.322	0.168	0.001	0.026

1) Case I. A vehicle to a CSP: For this case, wherein only the authentication of the session key between a vehicle and a CSP is considered, a traditional comparison method is used. This is because the other existing schemes [7], [13], [14] use this traditional method to compare computation overheads; thus, we also perform comparison using this method, and the results are summarized in Table IV. Because other existing schemes can be analyzed similarly, we analyze only the computational overhead of the proposed scheme. Table IV lists the computation overhead of each entity during the login, authentication and key agreement phases.

TABLE IV
Comparison on Computation Cost of Various Schemes in Case I

	Vehicle	RSU/FN/CSP	TA/CS/RA
Liu [13]	$T_{bp} + T_{bp.m} + 2T_{mtp} + 2T_{e/d}$ $\approx 43.906ms$	$T_{bp} + 2T_{mtp} + 2T_{e/d}$ $\approx 5.336ms$	$2T_{bp.m} + 3T_{mtp} + 3T_{e/d}$ $\approx 10.547ms$
Ma [14]	$3T_{W.m} + 4T_h$ $\approx 6.857ms$	$4T_{W.m} + 4T_h$ $\approx 1.292ms$	$10T_{W.m} + 11T_h$ $\approx 3.231ms$
Cui [7]	$3T_{W.m} + 8T_h$ $\approx 7.081ms$	$3T_{W.m} + 7T_h$ $\approx 0.973ms$	$2T_{W.m} + 10T_h$ $\approx 0.654ms$
Our	$3T_{M.m} + 9T_h$ $+T_{e/d} \approx 4.689ms$	$3T_{M.m} + 7T_h$ $\approx 0.511ms$	$2T_{M.m} + 10T_h$ $+T_{e/d} \approx 0.372ms$

RSU/FN/CSP denotes a Road Side Unit, a Fog Node or a CSP.
TA/CS/RA denotes a trusted entity.

For the proposed scheme, the computation cost of a vehicle from login to the completion of a request message requires two Montgomery curve point multiplication operations, four one-way hash function operations, and one symmetric encryption operation. In addition, only one Montgomery curve point multiplication operation and five one-way hash function operations are required during the key agreement phases. Thus

the execution time of a vehicle is $2T_{M.m} + 4T_h + T_{e/d} + T_{m.m} + 5T_h \approx 4.689$ ms. Meanwhile, the execution time of a CSP requires two Montgomery curve point multiplication operations and two one-way hash function operations during the broadcast phase before the vehicle login, and one Montgomery curve point multiplication operation and five one-way hash function operations during the authentication and key agreement phases. Thus, the computation cost of a CSP is $2T_{M.m} + 2T_h + 1T_{M.m} + 5T_h \approx 0.511$ ms. Because the RA does not participate in the login phase, we need to calculate only the overhead of the RA during the authentication and key agreement phases. The RA needs to conduct two Montgomery curve point multiplication operations, ten one-way hash function operations, and one symmetric decryption operation. Consequently, the execution time of the RA is $2T_{M.m} + 10T_h + T_{e/d} \approx 0.372$ ms.

2) Case II. n vehicles to m CSPs: In Case II, we consider the authentication of session keys between n vehicles and m CSPs. This case covers two other cases: wherein a vehicle applies to multiple CSPs simultaneously, and wherein a CSP is applied to by multiple vehicles simultaneously. For the sake of brevity, we have analyzed only the case of n vehicles to m CSPs. In Table V, we list the calculated computation overheads for n vehicles selecting from m CSPs during the login, authentication and key negotiation phases. We then introduce details of the analysis of the proposed scheme. The computation overheads of other existing schemes [7], [13], [14] can be obtained similarly.

TABLE V
COMPARISON ON COMPUTATION COST OF VARIOUS SCHEMES IN CASE II

	n vehicles	m RSUs/FNs/CSPs	TA/CS/RA
Liu [13]	$mnT_{bp} + mnT_{bp.m} + 2mnT_{mtp} + 2mnT_{e/d} \approx 43.906mn(ms)$	$mnT_{bp} + 2mnT_{mtp} + 2mnT_{e/d} \approx 5.336mn(ms)$	$2mnT_{bp.m} + 3mnT_{mtp} + 3mnT_{e/d} \approx 10.547mn(ms)$
Ma [14]	$3mnT_{W.m} + 4mnT_h \approx 6.857mn(ms)$	$4mnT_{W.m} + 4mnT_h \approx 1.292mn(ms)$	$10mnT_{W.m} + 11mnT_h \approx 3.231mn(ms)$
Cui [7]	$(m+2)nT_{W.m} + (5m+3)nT_h \approx 2.491mn + 4.59n(ms)$	$3mnT_{W.m} + 7mnT_h \approx 0.973mn(ms)$	$(mn+n)T_{W.m} + (3n+7mn)T_h \approx 0.329mn + 0.325n(ms)$
Our	$(m+2)nT_{M.m} + (5m+4)nT_h + nT_{e/d} \approx 1.621mn + 3.068n(ms)$	$(2m+mn)T_{M.m} + (2m+5mn)T_h \approx 0.173mn + 0.338m(ms)$	$(n+m)T_{M.m} + (2n+2m+6mn)T_h + nT_{e/d} \approx 0.006mn + 0.196n + 0.17m(ms)$

In our scheme, because a vehicle calculation request message contains the requested information for m CSPs, when n vehicles request m CSPs separately, the computation overhead from login to the completion of a request message requires only $n(2T_{M.m} + 4T_h + T_{e/d})$, instead of the computation overhead of a vehicle that requires nm times, as in the schemes [13] and [14]. Moreover, n vehicles receive the establishment key information from m CSPs, and thus each vehicle needs to calculate m shared secret keys; that is, the calculation overhead of n vehicles is $nm(T_{M.m} + 4T_h)$ during the key agreement phases. Thus, the execution time of n vehicles is $n(2T_{M.m} + 4T_h + T_{e/d}) + nm(T_{M.m} + 5T_h) \approx 1.621mn + 3.068n$ ms. Because our scheme uses the broadcast message method, the computation overhead of m CSPs in the broadcast phase is $m(2T_{M.m} + 2T_h)$. m CSPs need to calculate shared secrets for n vehicles separately, and thus, the computation overhead of m CSPs is $nm(T_{M.m} + 5T_h)$ during the key agreement phases. Therefore, the computation time of m CSPs is $2m(T_{M.m} + T_h) + nm(T_{M.m} + 5T_h) = 0.173mn + 0.338m$ ms. Because the RA receives only n request messages from n vehicles, the computational overhead for verifying the vehicle requires only $n(T_{M.m} + 2T_h + T_{e/d})$. Moreover, because of the additional cost of repeated calculations, the RA makes a record of CSPs that have been verified in the same period, and thus, the computational cost for verifying CSP is $m(T_{M.m} + 2T_h)$. Similarly, because the CSP does not know the real identity of each vehicle, RA generates unique authentication information for each CSP corresponding to each vehicle, and thus, the overhead for generating verification information is $6mnT_h$. Therefore, the computation time of the RA is $(n+m)T_{M.m} + (2n + 2m + 6mn)T_h + nT_{e/d} \approx 0.006mn + 0.196n + 0.17m$ ms for n vehicles selecting m CSPs .

According to Table IV, the total computational overhead of our proposed scheme for Case I is only 5.572ms, whereas those of the schemes by Liu et al. [13], Ma et al. [14], and Cui et al. [7] are 59.789ms, 11.38ms, and 8.708ms, respectively. Table V compares our proposed scheme with the related schemes [7], [13], [14]. The results demonstrate that the computational overhead increases with either the number of vehicles or CSPs. For $n = 1000$ and $m = 5$, that is, a case wherein 1000 vehicles apply to five CSPs simultaneously, the total computation times observed are 298.95s, 56.90s, 23.88s and 12.27s for [13], [14], [7], and our proposed scheme, respectively. Thus, for Case II, our proposed scheme expends less computation overhead than those of other existing schemes [7], [13], [14].

B. Comparison of Communication Overheads

Because the sizes of $\overline{q}$, $\tilde{q}$, and q are 192 bytes (1536 bits), 32 bytes (256 bits), and 32 bytes (256 bits) respectively, the sizes of the elements in G_1, G, and $\mathbb{G}$ are $192 \times 2 = 384$ bytes, $32 \times 2 = 64$ bytes, and $32 \times 2 = 64$ bytes, respectively. We also assume that the output size of a hash function, size of the timestamp, block size of the symmetric encryption/decryption, and size of a request are 32 bytes, 4 bytes, 16 bytes, and 4 bytes, respectively. Tables VI and VII compare the communication costs for two cases: one vehicle selecting one CSP and n vehicles selecting from m CSPs. We then analyze the communication overheads of these two cases. Because the analyses of the other existing schemes [7], [13], [14] are similar to the analysis of our proposed scheme, we discuss only our proposed scheme in the following subsection.

1) Case I. A vehicle and a CSP: The communication results for Case I are listed in Table VI. The proposed scheme requires four rounds in the login, authentication and key negotiation phases. In addition, the proposed scheme spends the following number of bytes in for four messages: $M_1 = \{W_i, PID_i, F_i\}$, $M_2 = \{N_r, W_i^{"}, GID_r, TS_r, \theta, \rho\}$, $M_3 = \{L_j, GID_r, \rho, TS_r, TS_{ij}\}$ and an anonymous message $M_j = \{R_j, PID_j, S_j, TS_j\}$ broadcasted by the CSP. Because W_i and R_j belong to $\mathbb{G}$; PID_i is the output of

TABLE VI
COMMUNICATION OVERHEAD COMPARISON IN CASE I

Scheme	No. of messages	Messages transmission among various entities	Size (bytes)
Liu et al. [13]	5	$V_i \xrightarrow{1172} RSU \xrightarrow{1556} TA \xrightarrow{1636} RSU \xrightarrow{1636} V_i \xrightarrow{384} V_j$	6384
Ma et al. [14]	4	$V_i \xrightarrow{132} FN \xrightarrow{328} CS \xrightarrow{260} FN \xrightarrow{228} V_i$	948
Cui et al. [7]	6	$V_i \xrightarrow{136} TA \xrightarrow{4} CSP \xrightarrow{136} TA \xrightarrow{196} CSP \xrightarrow{164} V_i \xrightarrow{32} CSP$	668
Our	4	$CSP \xrightarrow{132} V_i \xrightarrow{112} RA \xrightarrow{164} CSP \xrightarrow{136} V_i$	544

TABLE VII
COMMUNICATION OVERHEAD COMPARISON IN CASE II

Scheme	No. of messages	Messages transmission among various entities	Size (bytes)
Liu et al. [13]	5mn	$V_i \xrightarrow{1172mn} RSU \xrightarrow{1556mn} TA \xrightarrow{1636mn} RSU \xrightarrow{1636mn} V_i \xrightarrow{384mn} V_j$	6384mn
Ma et al. [14]	4mn	$V_i \xrightarrow{132mn} FN \xrightarrow{328mn} CS \xrightarrow{260mn} FN \xrightarrow{228mn} V_i$	948mn
Cui et al. [7]	5mn+n	$V_i \xrightarrow{136n} TA \xrightarrow{4mn} CSP \xrightarrow{136mn} TA \xrightarrow{196mn} CSP \xrightarrow{164mn} V_i \xrightarrow{32mn} CSP$	532mn+136n
Our	2mn+m+n	$CSP \xrightarrow{132m} V_i \xrightarrow{112n} RA \xrightarrow{164mn} CSP \xrightarrow{136mn} V_i$	300mn+132m+112n

the symmetric encryption; F_i, N_r, $W_i^{"}$, GID_r, θ, ρ, and L_j are the output of the hash function; and TS_j, TS_r, and TS_{ij} are the timestamps, the communication overheads of messages for our scheme are $\mid M_1 \mid = (64 + 16 + 32) = 112$ bytes, $\mid M_2 \mid = (32 + 32 + 32 + 4 + 32 + 32) = 164$ bytes, $\mid M_3 \mid = (32 + 32 + 32 + 4 + 4) = 136$ bytes and $\mid M_j \mid = (64 + 32 + 32 + 4) = 132$ bytes. Thus, for Case I, the cumulative communication overhead of our scheme is $(112 + 164 + 136 + 132) = 544$.

1) Case II. n vehicles and m CSPs: The communication results for Case II are listed in Table VII. For this case, our proposed scheme requires only $2mn + m + n$ rounds in the login, authentication and key negotiation phases. Because our scheme uses the broadcast message method on the CSP side, m CSPs need to broadcast only m anonymous messages $\{R_j, PID_j, S_j, TS_j\}$, where $j = 0, 1, \ldots, m-1$, instead of nm messages, as in the schemes [13], [14] and [7]. Because our scheme encrypts the anonymous information of the request CSP, even if n vehicles request m CSPs separately, they only need to send n pieces of request message M_1 instead of nm pieces. However, beyond that, because of the strong privacy of vehicle identity, the RA needs to generate unique authentication information for each vehicle$CCSP$ pair that wants to establish a secret private key, and thus, nm pieces of authentication message M_2 are sent by the RA. Similarly, m CSPs need to send shared secrets for n vehicles separately, and thus,nm pieces of authentication message M_3 are sent by the CSPs. Therefore, for Case II, the total communication overhead of our scheme is $m \mid M_j \mid + n \mid M_1 \mid + nm \mid M_2 \mid + nm \mid M_3 \mid = (132m + 112n + 164nm + 136nm) = 300mn + 132m + 112n$ bytes in Case II.

According to Table VI, the communication overhead of our proposed scheme for Case I is only 544 bytes, whereas those of the schemes by Liu et al. [13], Ma et al. [14], and Cui et al. [7] are 6384 bytes, 948 bytes, and 668 bytes, respectively. Meanwhile, Table VII compares the communication costs between our scheme and related schemes [7], [13], [14] for Case II. The results indicate that the communication overhead increases with either the number of vehicles or CSPs. Furthermore, for $n = 1000$ and $m = 5$, that is, a case wherein 1000 vehicles apply to five CSPs simultaneously, the cumulative communication overheads observed are 31171.88 KB, 4628.91 KB, 2730.47 KB and 1574.86 KB for [13], [14], [7], and our proposed scheme, respectively. Based on an analysis and comparison of Table VI and Table VII, we conclude that the communication cost of our proposed scheme is lower than those of the related schemes [7], [13], [14].

C. Comparison of Security and Functionality Features

Table VIII compares the security and functionality feature analyse of the related schemes [7], [13], [14] and our scheme. The symbol $\surd$ indicates that the scheme is secure or provides that feature. In contrast, the symbol $\times$ indicates that the scheme is insecure or does not provide that feature. This table indicates that only our proposed scheme can provide better security features than those of existing schemes [7], [13], [14].

TABLE VIII
COMPARISON OF SECURITY AND FUNCTIONALITY FEATURES

Security Features	[13]	[14]	[7]	Our
Provides Mutual Authentication	$\surd$	$\surd$	$\surd$	$\surd$
Provides Anonymity	$\surd$	$\surd$	$\surd$	$\surd$
Provides Traceability	$\surd$	$\surd$	$\surd$	$\surd$
Provides Un-linkability	$\surd$	$\surd$	$\surd$	$\surd$
Perfect Forward Secrecy	$\times$	$\surd$	$\surd$	$\surd$
Provision for revocation/re-registration	$\times$	$\times$	$\times$	$\surd$
Privileged Insider Attack	$\times$	$\times$	$\surd$	$\surd$
Offline Password Guessing Attack	$\surd$	$\times$	$\surd$	$\surd$
Impersonation Attack	$\times$	$\times$	$\times$	$\surd$
Replay Attack	$\surd$	$\times$	$\times$	$\surd$
Man-in-the-middle Attack	$\surd$	$\times$	$\times$	$\surd$
Ephemeral Secret Leakage Attack	$\times$	$\times$	$\times$	$\surd$

VI. CONCLUSION

This paper proposed a many-to-many authentication and key agreement scheme to achieve secure authentication between multiple vehicles and multiple CSPs for vehicular networks. In this scheme, the broadcast mechanism (at the CSP side) and hybrid encryption algorithm (such as elliptic curve, hash, and AES) are used to realize efficient many-to-many authentication. Moreover, the proposed scheme provides better SK-security compared with those of existing schemes [14] and [7],

even if the session ephemeral secret is unexpectedly leaked. Utilizing the widely-used ROR model and formal security analysis, the proposed scheme is proven to be resistant to several attacks. Finally, through performance evaluation, we conclude that the proposed scheme has lower computation and communication overhead, and provides higher security than those of existing schemes [7], [13], [14].

ACKNOWLEDGMENT

The work was supported by the National Natural Science Foundation of China (No. 61872001, No. 62011530046, No. U1936220), the Cooperation and Exchange Project between NSFC and RFBR (No. 20-57-53019, No. 62011530046), the Open Fund of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education (No. ESSCKF2018-03), the Open Fund for Discipline Construction, Institute of Physical Science and Information Technology, Anhui University and the Excellent Talent Project of Anhui University. The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

REFERENCES

[1] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 767–778, 2016.

[2] L. Zhang, "Otibaagka: a new security tool for cryptographic mix-zone establishment in vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2998–3010, 2017.

[3] 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.

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

[5] R. Yu, X. Huang, J. Kang, J. Ding, S. Maharjan, S. Gjessing, and Y. Zhang, "Cooperative resource management in cloud-enabled vehicular networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7938–7951, 2015.

[6] S. Azodolmolky, P. Wieder, and R. Yahyapour, "Cloud computing networking: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 54–62, 2013.

[7] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Transactions on Information Forensics and Security*, 2019.

[8] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "Pa-crt: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[9] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[10] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, "An extensible and effective anonymous batch authentication scheme for smart vehicular networks," *IEEE Internet of Things Journal*, 2020.

[11] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701–3709, 2017.

[12] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, and M. S. Obaidat, "A cooperative quality-aware service access system for social internet of vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2506–2517, 2017.

[13] 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.

[14] 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.

[15] A. I. Croce, G. Musolino, C. Rindone, and A. Vitetta, "Sustainable mobility and energy resources: A quantitative assessment of transport services with electrical vehicles," *Renewable and Sustainable Energy Reviews*, vol. 113, p. 109236, 2019.

[16] C. F. Daganzo and Y. Ouyang, "A general model of demand-responsive transportation services: From taxi to ridesharing to dial-a-ride," *Transportation Research Part B: Methodological*, vol. 126, pp. 213–224, 2019.

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

[18] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu, "Further observations on smart-card-based password-authenticated key agreement in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1767–1775, 2013.

[19] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014.

[20] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *IEEE transactions on wireless communications*, vol. 10, no. 2, pp. 431–436, 2010.

[21] M. Eltoweissy, S. Olariu, and M. Younis, "Towards autonomous vehicular clouds," in *International Conference on Ad hoc networks*. Springer, 2010, pp. 1–16.

[22] 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.

[23] S. Bhoi, S. Panda, S. Ray, R. Sethy, V. Sahoo, B. Sahu, S. Nayak, S. Panigrahi, R. Moharana, and P. Khilar, "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.

[24] R. I. Meneguette, A. Boukerche, and R. de Grande, "Smart: an efficient resource search and management scheme for vehicular cloud-connected system," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–6.

[25] J. Shao and G. Wei, "Secure outsourced computation in connected vehicular cloud computing," *IEEE Network*, vol. 32, no. 3, pp. 36–41, 2018.

[26] Z. Wei, H. Tang, F. R. Yu, M. Wang, and P. Mason, "Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4647–4658, 2014.

[27] 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.

[28] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, pp. 1–16, 2020.

[29] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2627–2637, 2017.

[30] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "Spacf: A secure privacy-preserving authentication scheme for vanet with cuckoo filter," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 283–10 295, 2017.

[31] Z. Shen and Q. Tong, "The security of cloud computing system enabled by trusted computing technology," in *2010 2nd International Conference on Signal Processing Systems*, vol. 2. IEEE, 2010, pp. V2–11.

[32] T. W. Chim, S.-M. Yiu, L. C. Hui, and V. O. Li, "Specs: Secure and privacy enhancing communications schemes for vanets," *Ad Hoc Networks*, vol. 9, no. 2, pp. 189–203, 2011.

[33] J. Benin, M. Nowatkowski, and H. Owen, "Framework to support per second shifts of pseudonyms in regional vanets," in *2010 IEEE 72nd Vehicular Technology Conference-Fall*. IEEE, 2010, pp. 1–5.

[34] Y.-Y. Chen, Y.-J. Wang, and J.-K. Jan, "The design of speedy seamless safe messaging mechanism in vanet," *International Journal of Computer Mathematics*, vol. 90, no. 12, pp. 2614–2630, 2013.

[35] P. Vijayakumar, M. Azees, A. Kannan, and L. J. Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015–1028, 2015.

[36] M. Azees, P. Vijayakumar, and L. J. Deboarh, "Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[37] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[38] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2001, pp. 453–474.

[39] ——, "Universally composable notions of key exchange and secure channels," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2002, pp. 337–351.

[40] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial internet of things," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[41] P. Sarkar, "A simple and generic construction of authenticated encryption with associated data," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 4, pp. 1–16, 2010.

[42] D. R. Stinson, "Some observations on the theory of cryptographic hash functions," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 259–277, 2006.

[43] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *International Workshop on Public Key Cryptography*. Springer, 2005, pp. 65–84.

[44] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 357–366, 2015.

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

[46] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs." *IACR Cryptol. ePrint Arch.*, vol. 2004, p. 332, 2004.

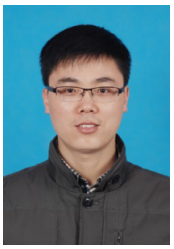
[47] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*. Springer, 1999, pp. 388–397.

[48] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[49] "Miracl cryptographic sdk," https://github.com/miracl/MIRACL/, accessed 29 Nov, 2019.

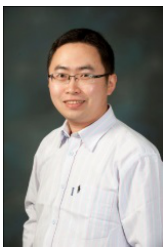
Jing Zhang is currently a PhD student in the School of Computer Science and Technology, Anhui University, Hefei, China. Her research interests include vehicular ad hoc network, IoT security and applied cryptography. She has nearly 20 scientific publications in reputable journals (e.g. IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Transportation Systems, Information Sciences, Science China Information Sciences and Vehicular Communications) and international conferences.

Hong Zhong was born in Anhui Province, China, in 1965. She received her Ph.D. degree in computer science from University of Science and Technology of China in 2005. She is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. Her research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN). She has over 120 scientific publications in reputable journals (e.g. IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Network and Service Management, IEEE Transactions on Big Data and IEEE Internet of Things Journal), academic books and international conferences.



Jie Cui was born in Henan Province, China, in 1980. He received his Ph.D. degree in University of Science and Technology of China in 2012. He is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. His current research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN). He has over 100 scientific publications in reputable journals (e.g. IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Computers, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Network and Service Management, IEEE Transactions on Emerging Topics in Computing, IEEE Transactions on Cloud Computing and IEEE Transactions on Multimedia), academic books and international conferences.

Yan Xu is an associate professor in the School of Computer Science and Technology, Anhui University. She received Ph.D. degree in University of Science and Technology of China in 2015. Her research interests include network and information security.



Lu Liu is the Professor of Informatics and Head of School of Informatics in the University of Leicester, UK. Prof Liu received the Ph.D. degree from University of Surrey, UK and MSc in Data Communication Systems from Brunel University, UK. Prof Liu's research interests are in areas of cloud computing, service computing, computer networks and peer-to-peer networking. He is a Fellow of British Computer Society (BCS).