Received December 17, 2018, accepted January 2, 2019, date of publication January 7, 2019, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891105

A Secure Authentication Protocol for Internet of Vehicles

CHIEN-MING CHEN[1], BIN XIANG[2], YINING LIU[3], AND KING-HANG WANG[4], (Member, IEEE)
[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Shandong 266590, China
[2]School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China
[3]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[4]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong

Corresponding author: King-Hang Wang (khwang0@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61662016 and in part by the Key Projects of Guangxi Natural Science Foundation under Grant 2018JJD170004.

ABSTRACT An Internet of Vehicles (IoV) allows forming a self-organized network and broadcasting messages for the vehicles on roads. However, as the data are transmitted in an insecure network, it is essential to use an authentication mechanism to protect the privacy of vehicle users. Recently, Ying *et al.* proposed an authentication protocol for IoV and claimed that the protocol could resist various attacks. Unfortunately, we discovered that their protocol suffered from an offline identity guessing attack, location spoofing attack, and replay attack, and consumed a considerable amount of time for authentication. To resolve these shortcomings, we propose an improved protocol. In addition, we provide a formal proof to the proposed protocol to demonstrate that our protocol is indeed secure. Compared with previous methods, the proposed protocol performs better in terms of security and performance.

INDEX TERMS Internet of Vehicles, authentication, anonymity, smart card.

I. INTRODUCTION

An Internet of Vehicles (IoV) allows for vehicles on roads to form a self-organized network. It provides multiple benefits, such as an in-built warning system that warns drivers of accidents so that they can decide quickly based on the provided roadside information. More sophisticated information can be possibly shared between vehicles and improve the safety and accuracy of auto-piloted vehicles. However, in the absence of an effective security and privacy measurement, an adversary can easily gather the transmitted data via networks that typically contain the private data of vehicle users. In addition to privacy concerns, data integrity or data authenticity is an important security topic in IoV. It might cause a tragedy if an adversary can generate fake messages to misguide a human driver or an auto pilot AI to make the wrong decisions. Those adversaries might attack the networks solely because of terrorism. They might also attempt to gain certain benefits from providing false information to other vehicles, such as attracting traffic to pass by their shops or competing parking lots.

Owing to IoV adversaries, it is essential to authenticate a vehicle before allowing it to join the network. A typical authentication scenario in a IoV is shown in Fig. 1. A vehicle V_i sends the join request to the trust authority (TA) with the assistance of a roadside unit (R_i). When the TA receives the request, it authenticates the vehicle and R_i and accepts joining request of V_i only if both R_i and V_i are legitimate.

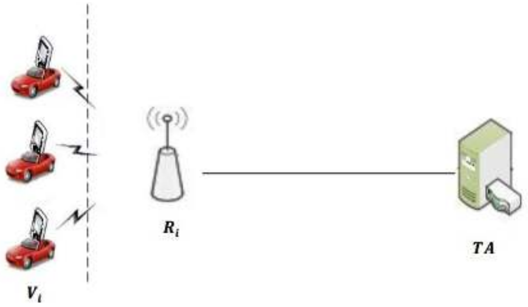

FIGURE 1. Typical authentication scenario of IoV.

Recently, many authentication protocols for IoV have been designed to protect the security of vehicles. In 2017, Ying *et al.* proposed an anonymous and lightweight authentication for secure vehicular networks called the ASC protocol [1]–[4]. They claimed that the ASC protocol could resist various attacks. However, we found that it is still vulnerable to offline identity guessing attacks and computation during authentication is time consuming. To address these shortcomings, we propose a new protocol to solve the above-mentioned problems. We also demonstrate that the proposed protocol is indeed secure with a formal proof. According to our performance analysis, the proposed protocol is more efficient compared with the previous related protocols.

The remainder of this paper is organized as follows. In Section II, we introduce state-of-the-art authentication protocols for the IoV. Subsequently, we review the ASC protocol in Section III and demonstrate that the ASC protocol is vulnerable to various attacks in Section IV. Next, we describe the assumptions and definitions that are required for our protocol. In Section VI, we present our proposed protocol. Sections VII and VIII further demonstrate that our proposed protocol is secure and efficient. Finally, we present the concluding remarks in Section IX.

II. RELATED WORK

Compared to traditional wireless networks, challenges in IoV involve more technical problems such as key distribution, privacy, mobility, incentives, bootstrap, and low tolerance for errors [5]. Currently, both industry and academia have introduced several approaches to protect the vehicular users' privacy and efficient authentication. The public key infrastructure (PKI) was first proposed to realize mutual authentication and key distribution among network vehicular users [6]–[14]. In 2005, Capkun and Hubaux [6] and Lazos *et al.* [7] each proposed an authentication protocol. In both protocols, the vehicles' location and public key signatures were used to prevent an attacker sitting on a roadside from claiming to be a legal vehicular user traveling on a highway. However, when considering large vehicle populations and limited storage requirements, the above location-based PKI security schemes are inefficient. Studier *et al.* [9], Lin *et al.* [10], and Ying *et al.* [11] proposed an authentication protocol based on hash chains. Unfortunately, these schemes did not provide vehicular user anonymity that could prevent attackers from obtaining sensitive information such as driver names and license plates. To reduce privacy leakage, anonymous authentication methods have been proposed to hide real identities [13], [14]. In [13] and [14], an ID is mapped to a distinct pseudo identity, and only the TA can retrieve the real identity from any pseudo identity.

Regarding large vehicle populations, the collection and storage of traffic-related data are highly challenging. To solve this, some approaches have been proposed to integrate cloud computing into vehicular networks such that the vehicles can share computation resources, storage resources, and bandwidth resources. Typically, a vehicular cloud, roadside cloud, and central cloud were included in these approaches [15]–[17]; however, each approach has its unique consideration. In Olariu *et al.*'s protocol [15], autonomous vehicular clouds were proposed to exploit underutilized resources. Bernstein *et al.* [16] proposed a platform-as-a-service cloud computing platform to solve cases where the clients are highly mobile, interactive, and functional. In Hussain *et al.*'s study [17], IoV clouds were divided into three architectural frameworks: vehicular clouds (VCs), vehicles using clouds (VuCs), and hybrid vehicular clouds (HVCs). Vehicles act as cloud service providers and clients in VCs and VuCs, respectively, and as both in HVCs.

Some scholars focused on the challenge of providing privacy to IoV users. The conditional privacy preserving authentication (CPPA) protocol was proposed to address this need. In the CPPA protocol, an attacker cannot obtain the real identity of a IoV user from the communication messages. Only trusted authority can identify the real identity of the IoV user from a given message that he/she has sent. This is to assure that the privacy of the IoV user is preserved while the system quickly revokes any dishonest user.

Shim [18] proposed a CPPA scheme using ID-based cryptography. A roadside unit (RSU) can authenticate the message in batch to reduce the computation overhead; in exchange, the trusted authority would require more time in handling vehicle revocation. Jianhong *et al.* [19] proposed another ID-based CPPA scheme that optimized the computation overheads and also supported batch authentication such as the study of Shim *et al.* However, it was reported that the scheme could not provide data integrity [20], [21]. Zhong *et al.* [4] recently proposed a CPPA scheme that reduces the vehicle revocation process and allows IoV users to modify their passwords more efficiently. Unfortunately, their scheme cannot provide unlinkability. An attacker of the system can match the vehicle registration time T_{OBU}^{reg} in the protocol to associate the owners of two messages.

Recently, some studies have suggested using smart cards to realize anonymous and lightweight authentication for secure vehicular networks [1], [3]. Wong *et al.* [1] provided an authentication protocol that imposes a light computational load owing to the usage of simple operations, such as the one-way hash function and XOR operations. In [2] and [3], to ensure strong security, Diffie-Hellman key exchange protocol was used when performing authentication. Unlike the literature above, with the usage of smart cards, extra parameters can be used as the building blocks of secure applications.

III. REVIEWS OF THE ASC PROTOCOL

This section briefly presents the ASC protocol for IoV. Three roles are present in the protocol. It is assumed that a single *TA* exists that manages all vehicles in the city. Many RSUs are managed by the *TA*. Each vehicular user (or driver) must be registered with the *TA* once and will be given a smart card. Users must insert the smart card into a vehicle before they can obtain a secure communication with the IoV. Their protocol consists of five phases: **the user registration phase** where each user registers with the *TA*; **the user login phase** where each user logs in to a vehicle with the smart card; **the user authentication phase** where a vehicle attempts to establish a secure communication session with the *TA*, **the data authentication phase** where secure datagrams are sent; **the password change phase** where the vehicular user attempts to change his/her passwords. We focus on **the user login phase**, **user authentication phase**, and **data authentication phase**

TABLE 1. Notations used herein.

Notation	Description
V_i	A vehicle i
R_i	the roadside unit i
x and y	the TA's secret key and public key
$H_i()$	the hash functions, $i = 0, 1, 2, 3$
$\oplus$	exclusive-or operation
$\|\|$	string concatenation

where severe security problems are found. The notations used in this section are listed in Table 1.

A. USER REGISTRATION PHASE

A vehicular is required to be registered before accessing the network for the first time. If a vehicular V_i desires to register with the *TA*, the following steps are performed.

Step 1: V_i selects its identity ID_{V_i} and password PW_i, and generates a random number n_i. Subsequently, V_i sends $\{ID_{V_i}, H_0(PW_i||n_i)\}$ to the *TA* through a secure channel. V_i will remember or record PW_i and n_i for the user login phase later.

Step 2: While receiving the registration request at time T_{reg}, the *TA* computes the following:

$$\begin{aligned} PVID_{V_i} &= H_0(ID_{V_i}) \\ A_i &= H_0(H_0(PW_i||n_i)||PVID_{V_i}) \\ N_i &= H_0(PW_i||n_i) \oplus H_0(x||PVID_{V_i}||T_{reg}). \end{aligned}$$

Step 3: The *TA* issues the data $\{A_i, N_i, g, p, y, H_0, H_1, H_2, H_3\}$ into a smart card and sends this card to V_i. It is noteworthy that x and y are the *TA*'s private key and public key, respectively, where $y = g^x \bmod p$. The computed values are also stored securely in the *TA*'s database.

B. USER LOGIN PHASE

V_i can perform this phase via the following steps to log in to a vehicle with the given smart card.

Step 1: V_i inserts the smart card into a vehicle and inputs ID_{V_i}, PW_i, and n_i.

Step 2: The smart card computes

$$A_i^* = H_0(H_0(PW_i||n_i)||H_0(ID_{V_i})),$$

and verifies it against the stored A_i. The smart card shall return a fail if the values are not matched.

C. USER AUTHENTICATION PHASE

When the vehicle attempts to establish a secure communication with a *TA*, the user authentication phase is invoked. As illustrated in Fig. 2, the steps can be explained as below:

Step 1: The smart card of V_i computes the following

$$\begin{aligned} k &= N_i \oplus H_0(PW_i||n_i) \\ C_1 &= y^{H_0(ID_{V_i})} \bmod p \\ DIDV_{i,j} &= H_0(C_1||H_0(ID_{V_i})||n_j) \\ CV_i &= H_0(DIDV_{i,j}||k) \end{aligned}$$

DIDV is called the dynamic login identity, where n_j is a random number and T_{V_i} is the current timestamp. Subsequently, it sends the login message $\{DIDV_{i,j}, CV_i, n_j, T_1\}$ to an RSU, R_i.

Step 2: When R_i receives the login message at time T_2, it aborts the protocol by verifying the timeout equation $T_2 - T_1 \geq \Delta T$. Otherwise, it computes

$$DIDR_i = DIDV_{i,j} \oplus H_0(ID_{R_i}).$$

It sends $\{DIDR_i, CV_i, n_j, T_2\}$ to the *TA*.

Step 3: On receiving the message from R_i at time T_3, the *TA* first verifies $T_3 - T_2 \leq \Delta T$ and aborts when it is timed out. Subsequently, it verifies the value of C_1 by

$$C_1 = y^{PVID_{V_i}} \bmod p. \quad (1)$$

Subsequently, it computes the following:

$$\begin{aligned} DIDV_{i,j}^* &= H_0(C_1||PVID_{V_i}||n_j) \\ H_0(ID_{R_i}^*) &= DIDV_{i,j}^* \oplus DIDR_i \\ k &= H_0(x||PVID_{V_i}||T_{reg}). \end{aligned}$$

Here, $H_0(ID_{R_i}^*)$ is used to validate the identity of R_i. It validates the value CV_i by $H_0(DIDV_{i,j}^*||k)$ and terminates the protocol if any of the above verification does not hold. [1] The *TA* subsequently computes the following:

$$\begin{aligned} C_3 &= H_1(PVID_{V_i}||k) \\ K_s &= H_1(C_3) \\ M_i &= E_{K_S}(K_c \oplus k). \end{aligned}$$

It sends $\{C_3, M_i, T_3\}$ to R_i.

Step 4: When R_i receives the message from the *TA* at time T_4, it verifies whether $T_4 - T_3 \leq \Delta T$, and aborts otherwise. Subsequently, it sends $\{C_3, M_i, T_4\}$ to V_i.

Step 5: On receiving the message from R_i at time T_5, V_i verifies whether $T_5 - T_4 \leq \Delta T$. It computes $C_3^* = H_1(H_0(ID_{V_i})||k)$ and validates C_3 against C_3^*. It terminates the protocol if any of the above verification fails. It computes $K_s^* = H_1(C_3)$ and uses it to decrypt M_i to obtain K_c. The value K_c will be served as a session key for further data transfer in the data authentication phase. The data authentication phase and password changing phase are not illustrated for simplicity.

IV. PROBLEMS WITH THE ASC PROTOCOL

Despite the ASC protocol exhibiting a better performance over the previous schemes [22]–[25], we found that the protocol does not scale up reasonably. Worst still, severe security problems occurred in the protocol despite a formal proof being provided in the original paper to prove security. We found that the protocol is insecure against an offline identity guessing attack [26], [27], a session linking attack,

[1] We found some problems in their original description that may hinder the readers from understanding the protocol.

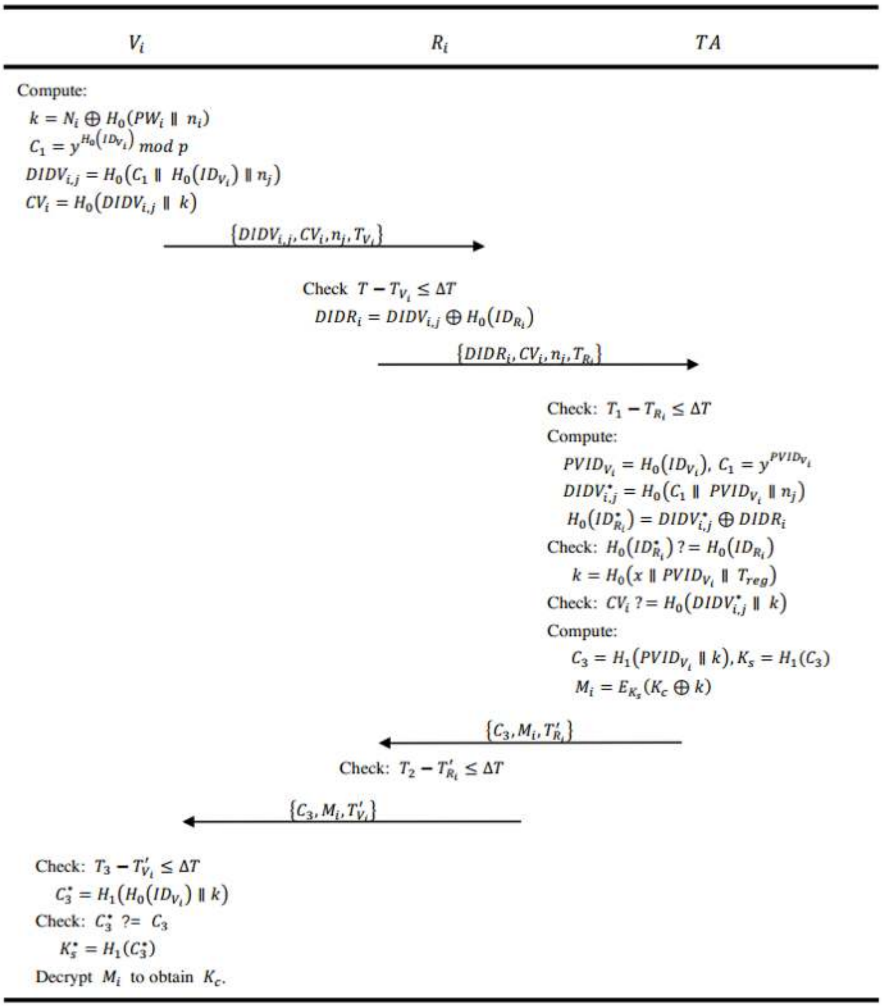
V_i
R_i
TA
Compute:
$k = N_i \oplus H_0(PW_i \parallel n_i)$
$C_1 = y^{H_0(ID_{V_i})} \bmod p$
$DIDV_{i,j} = H_0(C_1 \parallel H_0(ID_{V_i}) \parallel n_j)$
$CV_i = H_0(DIDV_{i,j} \parallel k)$
$\{DIDV_{i,j}, CV_i, n_j, T_{V_i}\}$
Check $T - T_{V_i} \le \Delta T$
$DIDR_i = DIDV_{i,j} \oplus H_0(ID_{R_i})$
$\{DIDR_i, CV_i, n_j, T_{R_i}\}$
Check: $T_1 - T_{R_i} \le \Delta T$
Compute:
$PVID_{V_i} = H_0(ID_{V_i}),\ C_1 = y^{PVID_{V_i}}$
$DIDV^*_{i,j} = H_0(C_1 \parallel PVID_{V_i} \parallel n_j)$
$H_0(ID^*_{R_i}) = DIDV^*_{i,j} \oplus DIDR_i$
Check: $H_0(ID^*_{R_i}) ?= H_0(ID_{R_i})$
$k = H_0(x \parallel PVID_{V_i} \parallel T_{reg})$
Check: $CV_i ?= H_0(DIDV^*_{i,j} \parallel k)$
Compute:
$C_3 = H_1(PVID_{V_i} \parallel k), K_s = H_1(C_3)$
$M_i = E_{K_s}(K_c \oplus k)$
$\{C_3, M_i, T'_{R_i}\}$
Check: $T_2 - T'_{R_i} \le \Delta T$
$\{C_3, M_i, T'_{V_i}\}$
Check: $T_3 - T'_{V_i} \le \Delta T$
$C^*_3 = H_1(H_0(ID_{V_i}) \parallel k)$
Check: C^*_3 ?= C_3
$K^*_s = H_1(C^*_3)$
Decrypt M_i to obtain K_c.

FIGURE 2. User authentication phase of Ying *et al.*'s protocol.

and a replay attack. The first two attacks allow for an attacker to break the anonymous property by linking two anonymous sessions of the same individual or even recovering a vehicle user's identity from an anonymous session. The third attack allows an attacker to impersonate a legitimate vehicle user without obtaining its smart card or password. In this section, we first demonstrate each of these problems, followed by identifying the fault of the formal proof given in the original paper to justify our claim.

A. SCALABILITY PROBLEM

In the user authentication phase of the ASC protocol, the TA is required to validate the message received with Eq. 1. However, this equation cannot be computed because $PVID_{V_i}$ is not sent over the network. Subsequently, computing $DIDV_{i,j}$ is also problematic. The only possible method for the TA to perform this protocol is to iterate every possible registered user V_κ and compute the following:

$$PVID_{V_\kappa} = H_0(ID_{V_\kappa}) \tag{2}$$

$$C_\kappa = y^{PVID_{V_\kappa}} \mod p \tag{3}$$

$$DIDV_{\kappa,j} = H_0(C_\kappa || PVID_{V_\kappa} || n_j). \tag{4}$$

Subsequently, the TA can confirm the identity of V_κ if $DIDV_{\kappa,j}$ equals $H_0(ID_\kappa) \oplus DIDR_i$. Otherwise, the TA must use another registered user. Even with pre-computation, only Eq. 2 and 3 can be pre-computed while Eq. 4 depends on a large random value n_j that must be computed on the spot. In most IoV applications that require fast response and support of a large number of users, fetching data intensively and computing a number of users are impractical. The time required for one hash computation was measured as 0.00074 ms [2]. Assume that we need to support one million users; therefore, an average of 0.37 s is required to identify the identity of a user, and 0.74 s to reject an attack message. This number does not include the times required for fetching the pre-computed data from the database and the concurrent authentication protocol request from other vehicles. This can allow for a denial-of-service (DoS) where a massive number of fake messages cannot be rejected at an earlier stage of the protocol, thus resulting in the exhaust of the TA's resources.

B. LOCATION SPOOFING ATTACK

A location spoofing attack suggests that an attacker is attempting to falsify its location and convince the server to believe it. The ASC protocol is not designed to handle location spoofing attacks and it is rather vulnerable against location spoofing. There is no secure and authenticated channel assumed between an R_i and the TA. It is possible for an attacker to intercept the message sent from V_i in the first round of the protocol $DIDV_{i,j}$ to a particular RSU, R_j, and also the message from R_j to the server such that the attacker can deduce $H_o(ID_{R_j}) = DIDR_j \oplus DIDV_{i,j}$. To falsify its location later, the attacker can compute a valid $DIDR_j$ with the $H_o(ID_{R_j})$ that was obtained previously. The server will be misled when the attacker is logging in near the RSU, R_j.

C. OFFLINE IDENTITY GUESSING ATTACK

The ASC protocol is claimed to be anonymous. We found that an attacker adversary $\mathcal{A}$ can launch an offline identity guessing attack as follows. Similar to the password guessing attack where an $\mathcal{A}$ contains a list of possible passwords, an identity guessing attacker $\mathcal{A}$ contains a list of user IDs. We believe that this assumption is reasonable because the user ID is chosen by users, and certain patterns of ID are favored by human users. Additionally, it is not explicitly assumed that the user ID is secret information in this protocol. In some other applications such as a discussion forum or p2p network, user IDs are actually published. The attacker $\mathcal{A}$ starts the attack by capturing the transmitted data $\{DIDV_{i,j}, CV_i, n_j, T_1\}$ in the user authentication phase and performs Algorithm 1.

Algorithm 1 Offline Identity Guessing Attack

```
procedure Guessing(User list: $\mathcal{L}$, Captured Message: $\{DIDV_{i,j}, n_j, \}$)
    for $V_\kappa \leftarrow \mathcal{L}$ do
        $C_\kappa = y^{H_0(ID_{V_\kappa})} \mod p$
        $DIDV_{\kappa,j} = H_0(C_\kappa||H_0(ID_{V_\kappa})||n_j)$
        if $DIDV_{\kappa,j} = DIDV_{i,j}$ then
            return $V_\kappa$.
        end if
    end for
    return Fail.
end procedure
```

With the steps above, the $\mathcal{A}$ can successfully guess the identity of V_i with complexity $\mathcal{O}(u)$, where u is the number of users. The function returns a fail if either V_i is not in the user list or the capture message is invalid. The function returns an incorrect result only if collision appears in the l-bit hash function H_0, with probability bounded by $1/2^{l-1}$.

D. SESSION-LINKING ATTACK

Assume that the user ID is well protected such that the attacker cannot obtain a list of users; therefore, the protocol is still insecure because two anonymous sessions can be linked together if they are associated with the same V_i. We observed that the value C_3 returned from the TA in the user authentication phase equals to $H_1(PVID_{V_i}||k)$, where $PVID_{V_i}$ and k are fixed values derived from ID_{V_i}, private key of the TA x, and time of registration T_{reg}. Thus, C_3 remains constant for the same user every time he/she attempts to initiate a session with the TA. By observing the value of C_3, the $\mathcal{A}$ can conclude a set of sessions belonging to the same user where further vehicle/user tracking is possible.

E. REPLAY ATTACK

The $\mathcal{A}$ can impersonate a legitimate user without learning its ID and password, and without obtaining the smart card by launching a replay attack. The protocol is designed to be replay–attack proof by facilitating a timestamp in each step of the protocol, such that replaying a message earlier than ΔT will be rejected by R_i. However, the timestamp is separated from the other part of the message while no other authenticator contains the timestamp. Therefore, once the $\mathcal{A}$ has captured a message $\{DIDV_{i,j}, CV_i, n_j, T_1\}$ in the air, the $\mathcal{A}$ can impersonate V_i by sending $\{DIDV_{i,j}, CV_i, n_j, T_1'\}$, where T_1' is the time that an $\mathcal{A}$ replays the message.

F. STOLEN SMART-CARD ATTACK

The authors of ASC claimed that losing a smart card will not allow users to generate a valid login message $DIDV_{i,j} = H_0(C_1||H_0(ID_{V_i})||n_j)$, even if A_i, N_i is extracted by the attacker. However, by conducting an offline password guessing attacks against the value A_i with $H_0(H_0(ID_{V_\kappa})||H_0(PW_\kappa||n_i))$, the attacker can easily recover the password and impersonate the victim.

G. PROBLEM WITH THE PROOF

A formal proof was written in the previous work to argue the security of the ASC protocol. The probability for an attacker to differentiate a session key against the same-length random string was claimed to be bounded by $1/2 + \epsilon$, where ϵ is a negligible function of the numbers of send queries, encryption queries, hash queries allowed by the attacker, and the length of the hash functions. The proof does not include the anonymous property of the protocol. For the replay attack, it somehow contradicts with the result of the proof. We are not suggesting any new findings regarding the incapability of a formal proof in proving the security of a cryptographic protocol. Instead, we would like to indicate the fault of the proof given in the ASC protocol.

The proof is rather complicated, and we can only provide a high-level pointer to where we suggest the error is. If we look closer at Game G_4 of the proof, the proof simulator aborts the instance when C_1 and C_3, K_s are not found in $\Lambda\Gamma_\psi$, i.e., the previous output of the simulator. Indeed this game change is not perfectly indistinguishable to an attacker as it is claimed, even if **AskPara**$_4$ does not occur. Consider the case that the protocol is executed for the first time; $\Lambda\Gamma_\psi$ is empty. The simulator of G_4 will therefore reject any instance regardless of whether it is legitimate. This fault exposes the loophole that a replay message described in our attack will be excluded wrongly in the protocol proof, which is unpreventable.

V. ASSUMPTION AND DEFINITION

Before we present our protocol, we state the assumptions and definitions that are required for our protocol.

A. ASSUMPTION

We assume a similar setting in ASC - a single *TA* and some RSUs (denoted as $\mathcal{R} = \{R_1, R_2, \cdots\}$) exist. Some registered vehicles (denoted as $\mathcal{V} = \{V_1, V_2, \cdots\}$) attempt to communicate anonymously with the *TA* through these RSUs. A vehicle must undergo a registration process with the *TA* and will receive a smart card that contains some authentication secrets.

In addition, we assume that a secure and authentic communication channel exists between an RSU and *TA*. This assumption is practical when an RSU is implanted into a traffic light or a road light. A physical intrusion on these units can become easily noticeable by law enforcers. Each of these units contains a unique pair-wise key shared with the *TA*. All communications between the unit and *TA* will be encrypted and padded with a message authentication code to ensure its security and authenticity.

Furthermore, we eliminate the use of passwords in this protocol owing to its practicality. The primary purpose of having a password in this setting is to guard against the smart card from being accessed by an attacker without the smart card (vehicle) owner's explicit consent. Nevertheless, a good physical vehicle security (i.e., anti-thief security system) should have already addressed this issue and this password guard would be redundant. Additionally, most of the vehicles on the market today are not equipped with good input devices for passwords. Even in vehicles with a larger touch screen interface (e.g., a Tesla Model S), it is still not designed for a completely confidential password input process. In this study, we assume that the attackers can retrieve the secrets inside the smart cards of some vehicles owned by the attackers.

B. DEFINITION

We prove our propose protocol using formal security notions based on the study of Bellare *et al.* [28]. It is described as a *game* played by an attacker who claims that he/she can break our protocol. The input of this game is an instance of a cryptographic hard problem. At the end of the game, the attacker will be challenged to answer a question (e.g., to differentiate the session key against a random string). In our proof, we demonstrate that the answer provided by the attacker will enable the game to solve the cryptographic hard problem with a non-negligible probability. This implies that breaking the protocol would lead to a solution to a hard problem. As the hard problem is assumed to be unsolvable, the opposite would therefore suggest that our protocol is unbreakable.

Definition 1 (Computational Diffie–Hellman (CDH) Problem): Let $\mathcal{G}$ be a multiplicative group and g a generator of $\mathcal{G}$. Given a CDH instance $x = g^a, y = g^b, g, \mathcal{G}$, the problem is to compute g^{ab}.

Definition 2 (Intractable): A problem is regarded as intractable if no polynomial time algorithm can solve this problem with non-negligible probability.

Assumption 1: It is assumed that the CDH problem is intractable for some $\mathcal{G}$, g.

We define a set of queries that an $\mathcal{A}$ can issue when the protocol is executed. These queries are to model the active/passive attacks of an $\mathcal{A}$.

1) $send(A, S, \texttt{Step}, M)$ is a query indicating that a message M is sent to a protocol participant A in the communication session S where $\texttt{Step}$ is an integer that refers to the particular step of the protocol. This models an active attacker attempting to send a message to A. A will either continue with the protocol, accept it (only if it is the last step of the protocol) or reject it. The communication session S can be understood as a TCP connection, for instance.
   The query is overloaded in the following format: $send(A, S, 0, B)$ for some A, B. This refers to the case where the $\mathcal{A}$ requests participant A to start the protocol with another participant B.
2) $reveal(S)$ is a query to model where the $\mathcal{A}$ can obtain the session key from a communication session S, perhaps from the misuse of the session key or a device hack. This query can only be issued to a session that terminates at an accepted state.
3) $test(S)$ is a special query where a binary random number c will be generated. If $c = 0$, the session key of the accepted session S will be returned. When $c = 1$, a random string with the same length and distribution as the session key will be returned. The attacker $\mathcal{A}$ must guess the value of c. It is noteworthy that this query can only be applied to a session that is in an accepted state and also not been revealed.

When the protocol is executed, an attacker $\mathcal{A}$ can perform the above queries polynomial times. At the end, the attacker is required to issue a test query and output its guess c' on the value of c.

Definition 3 (Advantage): Let **Suc** be the event where $c' = c$; we denote the advantage of the adversary in breaking the protocol P as

$$Ad_P^{se}(A) = 2\Pr(\mathbf{Suc}) - 1$$

Definition 4 (Security): We regard the protocol P as secure if no polynomial-time attacker $\mathcal{A}$ can have a non-negligible advantage, i.e.,

$$\forall \mathcal{A}\, \exists n_c, \quad Ad_P^{se}(A) < ke^{-n_c}$$

VI. THE PROPOSED PROTOCOL

The protocol requires every vehicular user to register with the *TA* once. It is assumed that this registration process can be performed securely without any attacker interference. Nevertheless, an attacker can also register with the *TA* using one or more identities.

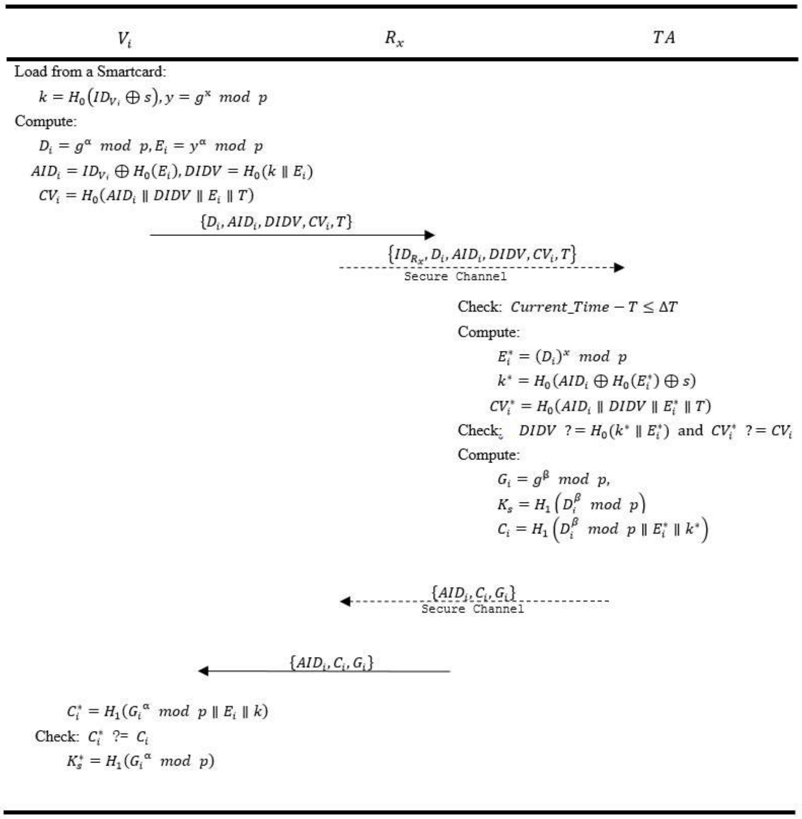
V_i
R_x
TA
Load from a Smartcard:
$k = H_0(ID_{V_i} \oplus s), y = g^x \ mod \ p$
Compute:
$D_i = g^\alpha \ mod \ p, E_i = y^\alpha \ mod \ p$
$AID_i = ID_{V_i} \oplus H_0(E_i), DIDV = H_0(k \parallel E_i)$
$CV_i = H_0(AID_i \parallel DIDV \parallel E_i \parallel T)$
$\{D_i, AID_i, DIDV, CV_i, T\}$
$\{ID_{R_x}, D_i, AID_i, DIDV, CV_i, T\}$
Secure Channel
Check: $Current\_Time - T \leq \Delta T$
Compute:
$E_i^* = (D_i)^x \ mod \ p$
$k^* = H_0(AID_i \oplus H_0(E_i^*) \oplus s)$
$CV_i^* = H_0(AID_i \parallel DIDV \parallel E_i^* \parallel T)$
Check: $DIDV \ ?= H_0(k^* \parallel E_i^*)$ and $CV_i^* \ ?= CV_i$
Compute:
$G_i = g^\beta \ mod \ p,$
$K_s = H_1(D_i^\beta \ mod \ p)$
$C_i = H_1(D_i^\beta \ mod \ p \parallel E_i^* \parallel k^*)$
$\{AID_i, C_i, G_i\}$
Secure Channel
$\{AID_i, C_i, G_i\}$
$C_i^* = H_1(G_i^\alpha \ mod \ p \parallel E_i \parallel k)$
Check: $C_i^* \ ?= C_i$
$K_s^* = H_1(G_i^\alpha \ mod \ p)$

FIGURE 3. User authentication phase of the proposed protocol.

The *TA* picks a set of system parameters. $\mathbf{F}_p$ is a prime field where the CDH assumption holds. g is a generator of the field. $x \in \mathbf{Z}_p$ is a private key and $y = g^x \mod p$ is a public key. s is a server secret. H_0 and H_1 are two secure one-way hash functions.

A vehicular user V_i chooses the identity ID_{V_i} that has never been registered by other users and sends it to the *TA*. When receiving the registration message, the *TA* first verifies if the identity has not yet been registered from its database. Subsequently, it computes $k = H_0(ID_{V_i} \oplus s)$. The *TA* issues a smart card to V_i. The smart card contains $\{ID_{V_i}, k, g, p, y, H_0, H_1\}$. V_i should insert the smart card in its vehicle. Throughout this process, the communication is assumed confidential and authentic.

For each time a vehicular user V_i wants to connect to the *TA* through a nearby RSU R_x, it executes the following steps to obtain a session key. The details are illustrated in Fig. 3.

Step 1: V_i retrieves k and y from the smart card and generates a random integer α, and computes the following using the current time T:

$$\begin{aligned} D_i &= g^\alpha \mod p, \\ E_i &= y^\alpha \mod p, \\ AID_i &= ID_{V_i} \oplus H_0(E_i), \\ DIDV &= H_0(k||E_i), \\ CV_i &= H_0(AID_i||DIDV||E_i||T). \end{aligned}$$

Subsequently, V_i sends the message $\{D_i, AID_i, DIDV, CV_i, T\}$ to an RSU R_x.

Step 2: When R_x receives the login message from a vehicle, it drops the message if T in the login message expires. Otherwise, it appends its ID ID_{R_x} to the message and relays it to the *TA*.

Step 3: On receiving the message from R_x, the *TA* first verifies whether T in the message has expired, i.e., $Current\_Time - T \leq \Delta T$. It subsequently tries to validate the login message using the following equations:

$$\begin{aligned} E_i^* &= (D_i)^x \mod p, \\ k^* &= H_0(AID_i \oplus H_0(E_i^*) \oplus s), \\ CV_i^* &= H_0(AID_i||DIDV||E_i^*||T). \end{aligned}$$

The authentication fails if $DIDV \neq H_0(k^*||E_i^*)$ or $CV_i^* \neq CV_i$. Otherwise, it accepts the login request and records that V_i connects via R_x at time T.

It subsequently generates a random integer β. It uses them to compute the following:

$$\begin{aligned} G_i &= g^\beta \mod p, \\ K_S &= H_1(D_i^\beta \mod p), \\ C_i &= H_1(D_i^\beta \mod p||E_i^*||k^*). \end{aligned}$$

It sends $\{AID_i, C_i, G_i\}$ to R_x and the RSUs near R_x through a secure channel. R_x and these RSUs broadcast the same message on air. The messages are also sent to the nearby RSU to prevent V_i moving away from R_x.

Step 4: V_i performs the following when the broadcast message begins with AID_i. It computes the following:

$$C_i^* = H_1(G_i^{\alpha} \bmod p||E_i||k),$$
$$K_S^* = H_1(G_i^{\alpha} \bmod p).$$

It accepts the protocol if $C_i^* = C_i$ using K_S as the session key.

VII. SECURITY ANALYSIS OF THE PATCHED PROTOCOL

We prove that the protocol can provide mutual authentication by contradiction. If an attacker $\mathcal{A}$ can break the security of the protocol, or more precisely $Ad_P^{se}(A)$ is non-negligible, we can formulate a (t, ϵ)-CDH attacker Δ running in time t such that

$$\mathbf{Suc}^{cdh}(\Delta) = \Pr(\Delta(g^a, g^b, g, \mathcal{G}) = g^{ab}) \geq \epsilon$$

where a and b are random integers, and g is a generator in a finite cyclic group $\mathcal{G}$. Because the CDH-assumption holds for all polynomial t and any non-negligible ϵ, by contradiction no such attacker exists that can break our protocol.

Theorem 1: Given a finite cyclic group $\mathcal{G}$ of size p, the group's generator g, a password dictionary of size $|D|$, and our proposed protocol P, an attacker $\mathcal{A}$ can break our protocol with resource (t, q_s, q_h, q_e), and has an advantage bounded by the following:

$$Ad_P^{se}(A) \leq\leq 2(q_s \cdot q_h)\mathbf{Suc}^{cdh}(\mathbf{G}_7) + \frac{2q_s^2+3q_h+(9q_s+q_h)^2}{2^{l-1}}$$

where t is the running time, l is the length of the hash and encryption, q_s is the number of send queries, and q_h is the number of hash queries that the attacker can make.

Proof: We define a series of games $\mathbf{G}_n$, where $n = 0, 1, 2, \ldots,$ starting with $\mathbf{G}_0$ as a faithful simulation of our protocol. Each subsequent game modifies some subtle details from a previous game such that the attacker can distinguish the difference between two games with little probability. The series of games ends with $\mathbf{G}_7$ such that the advantage of the $\mathcal{A}$ is negligible. The game simulator uses the input of a CDH instance $(g^a, g^b, g, \mathcal{G})$ and interacts with the attacker $\mathcal{A}$. We also define the following event when playing $\mathbf{G}_n$:

- $\mathbf{Suc}_n$ occurs if the $\mathcal{A}$ correctly guesses the value of c in the test query.
- $\mathbf{HashCDH} - \mathbf{0}_n(\lambda)$ occurs if the $\mathcal{A}$ has to make an H_0 query that contains $(g^{ab})^{\lambda}$ in the input.
- $\mathbf{HashCDH} - \mathbf{1}_n(\lambda)$ occurs if $\mathcal{A}$ has to make an H_1 query that contains $(g^{ab})^{\lambda}$ in the input.

We describe the details of the games as below.

- Game $\mathbf{G}_0$: This game executes the protocol described in the earlier session. Thus, by definition,

$$Ad_P^{se}(A) = 2\Pr(Suc_0) - 1$$

- Game $\mathbf{G}_1$: In this game, we simulate our hash functions H_0 and H_1 by maintaining two private lists: $\Lambda\Gamma_0$ and $\Lambda\Gamma_1$ on every hash query to H_0 and H_1, respectively. When a query exists, either by the simulator itself or from the attacker, the simulator searches the input from the list and replies the record if it is found. Otherwise, it generates a random output with the same probability distribution of the hash and adds the record to the list. This change is indistinguishable by the attacker unless a collision appears in the hash or encryption. By birthday paradox,

$$|\Pr(\mathbf{Suc}_1) - \Pr(\mathbf{Suc}_0)| \leq \frac{(9q_s + q_h)^2}{2^l}.$$

It is noteworthy that nine distinct hash queries are invoked at most in one protocol execution.

- Game $\mathbf{G}_2$: $\mathbf{G}_2$ is modified from $\mathbf{G}_1$ such that when a user is registered with the *TA*, the ID, k and a flag that indicates if this ID is registered by the attacker will be recorded in a private table $\Lambda\Gamma_2$. In Step 3, the simulator performs an additional verification: it computes $ID^* = AID_i \oplus H_0(E_i^*)$ and rejects the login request if $ID^* \notin \Lambda\Gamma_2$. This change is indistinguishable by the $\mathcal{A}$ unless $H_0(AID_i \oplus H_0(E_i^*) \oplus s)$ has been queried by the $\mathcal{A}$. It is noteworthy that the $\mathcal{A}$ does not exhibit the value of s and s is never sent. Therefore,

$$|\Pr(\mathbf{Suc}_2) - \Pr(\mathbf{Suc}_1)| \leq \frac{q_h}{2^l}.$$

- Game $\mathbf{G}_3$: $\mathbf{G}_3$ is modified from $\mathbf{G}_2$ such that k is not generated by a hash function. Instead, when a user is registered with the simulator, k is assigned to a random $\log_2 p$-bit string. In Step 3, when k^* is required, the simulator computes $ID^* = AID_i \oplus H_0(E_i^*)$ and retrieves k^* from $\Lambda\Gamma_2$. This change is indistinguishable by the attacker.

$$\Pr(\mathbf{Suc}_3) = \Pr(\mathbf{Suc}_2).$$

Up to this game, we have removed the role of s.

- Game $\mathbf{G}_4$: $\mathbf{G}_4$ is modified from $\mathbf{G}_3$ such that instead of computing $DIDV$ from k and E_i, $DIDV$ is assigned by a $\log_2 p$-bit long random string and record $(k, D_i, E_i, DIDV)$ in a private list $\Lambda\Gamma_3$. In Step 3, it verifies if ID^* is registered by the attacker. If it does, it looks up the value of $DIDV$ from $\Lambda\Gamma_0$ for the hash $H_0(k^*||E_i)$ that must have been queried by the $\mathcal{A}$ before. If ID^* is not registered by the $\mathcal{A}$, implying that the attacker does not contain k, the simulator retrieves the value of $DIDV$ from $\Lambda\Gamma_3$. The simulator rejects if $DIDV \notin \Lambda\Gamma_3$ and ID^* is not registered by the $\mathcal{A}$.
From the perspective of the $\mathcal{A}$, $\mathbf{G}_4$ is indistinguishable from $\mathbf{G}_3$ unless either 1) the same D_i has been generated by the same V_i or 2) $\mathcal{A}$ has queried $H_0(k||E_i)$. Thus,

$$|\Pr(\mathbf{Suc}_4) - \Pr(\mathbf{Suc}_3)| \leq \overbrace{\frac{q_s^2}{2^l}}^{\texttt{Case 1}} + \overbrace{\frac{q_h}{2^l}}^{\texttt{Case 2}}.$$

TABLE 2. Comparison of security requirements against related works.

Protocols	Ying et al. [3]	ASC [2]	Zhong et al. [4]	Our Protocol
R1:user anonymity	Y	Y	Y	Y
R2:insider attack	N	Y	Y	Y
R3:stolen smart card attack	Y	Y	Y	Y
R4:offline password guessing attack	Y	Y	Y	Y
R5: replay attack	Y	Y	Y	Y
R6: user impersonation attack	Y	Y	Y	Y
R7: RS impersonation attack	Y	Y	Y	Y
R8: TA impersonation attack	Y	Y	Y	Y
R9: DoS attack	N	N	Y	Y

- Game $\mathbf{G}_5$: Modified from $\mathbf{G}_4$, In Step 3, if ID^* is not registered by the $\mathcal{A}$, the simulator computes $C_i = H_1(D_i^\beta \bmod p||X||k^*)$ for a random $\log_2 p$-bit string X that will be reused in other sessions. In Step 4, the simulator computes $C_i^* = H_1(G_i^\alpha \bmod p||X||k)$. It is noteworthy that k is not known to the $\mathcal{A}$ and is never sent without a hash. Furthermore, it rejects the authentication if G_i has been modified by the $\mathcal{A}$.
Consequently, $\mathbf{G}_5$ is indistinguishable to $\mathbf{G}_4$ by the $\mathcal{A}$ unless either 1) the same D_i^β is used in some other sessions or 2) the $\mathcal{A}$ has successfully guessed the value of X. Thus,
$$|\Pr(\mathbf{Suc}_5) - \Pr(\mathbf{Suc}_4)| \leq \overbrace{\frac{q_s^2}{2^l}}^{\texttt{Case 1}} + \overbrace{\frac{q_h}{2^l}}^{\texttt{Case 2}}.$$
- Game $\mathbf{G}_6$: The simulator substitutes its public key $y = g^x \bmod p$ by the CDH instance $y = g^a \bmod p$. In Step 1, we change the computation of D_i and E_i as
$$D_i = (g^b)^\alpha \bmod p,$$
$$E_i = X^\alpha.$$
In Step 4, we change the computation of C_i^* and K_i^* as follows:
$$C_i^* = H_1(D_i^\beta \bmod p||X||k),$$
$$K_i^* = H_1(D_i^\beta \bmod p).$$
Recall that $g^b \bmod p$ the CDH instance and α, β are random numbers generated in this session.
The simulator also records (D_i, E_i, α) into a private table $\Lambda\Gamma_4$. In Step 3, E_i^* is retrieved from $\Lambda\Gamma_4$ by looking up D_i. This change is indistinguishable by the $\mathcal{A}$ unless the following events occurred:
  1) $\mathcal{A}$ has queried $H_0(g^{ab\alpha})$; or
  2) $\mathcal{A}$ has queried $H_0(AID_i||DIDV||g^{ab\alpha}||T)$.

  These two events imply $\mathbf{HashCDH} - \mathbf{0}_6(\alpha)$. It is noteworthy that the statistical distribution of D_i and G_i remain unchanged. Thus,
$$|\Pr(\mathbf{Suc}_6) - \Pr(\mathbf{Suc}_5)| \leq \Pr(\mathbf{HashCDH} - \mathbf{0}_6(\alpha))$$
- Game $\mathbf{G}_7$: In this game, we change the G_i sent from Step 3 to $G_i = g^{a\beta} \bmod p$, and K_s will be assigned randomly. This change is indistinguishable to the attacker unless the attacker has queried $H_1(g^{ab\alpha\beta} \bmod p)$, which can be denoted as $\mathbf{HashCDH} - \mathbf{1}_7(\alpha\beta)$.
$$|\Pr(\mathbf{Suc}_7) - \Pr(\mathbf{Suc}_6)| \leq \Pr(\mathbf{HashCDH} - \mathbf{1}_7(\alpha\beta))$$
Because K_s is a random number and is irrelevant to the value of the coin c,
$$\Pr(\mathbf{Suc}_7) = 1/2$$

TABLE 3. Time cost of each operation executed on an embedded device (in ms).

Operation	Notation	Time cost
Modular exponentiation	T_e	4.76ms
Multiplication of points on ECC(secp256r1)	T_m	20.23ms
Hash(SHA-256)	T_h	0.03ms
Symmetric encryption/decryption	T_{enc}	0.12ms

At the end of the game, the simulator will pick one of the hash queries out of the q_h queries made by the $\mathcal{A}$ and guesses it be that corresponding to the event $\mathbf{HashCDH} - \mathbf{0}_6(\alpha)$ or $\mathbf{HashCDH} - \mathbf{1}_7(\alpha\beta)$. If an H_0 query is selected, e.g., $\mathcal{A}$ queried $H_0(\gamma)$ or $H_0(AID_i||DIDV||\gamma||T)$, the simulator randomly picks one α used among q_s sessions and computes $r = \gamma^{\alpha^{-1}} \bmod p$. In case a H_1 query is selected, e.g., $\mathcal{A}$ queried $H_1(\gamma)$, the simulator randomly picks one pair of α, β used among q_s sessions and computes $r = \gamma^{(\alpha\beta)^{-1}}$. It returns r as the answer of $g^{ab} \bmod p$ to the CDH instance. By definition, we have $\mathbf{Suc}^{cdh}(\mathbf{G}_7) = \Pr(r = g^{ab})$. Additionally,
$$\begin{aligned}\Pr(r = g^{ab}|\mathbf{HashCDH} - \mathbf{1}_7(\alpha\beta) \cup \mathbf{HashCDH} - \mathbf{0}_6(\alpha))\\ = \frac{1}{q_s \cdot q_h}\Pr(\mathbf{HashCDH} - \mathbf{1}_7(\alpha\beta) \cup \mathbf{HashCDH} - \mathbf{0}_6(\alpha))\\ \leq (q_s \cdot q_h)\Pr(r = g^{ab}).\end{aligned}$$

By chaining up the games and the equation above, we have
$$\begin{aligned}&\Pr(\mathbf{Suc}_0)\\ &\leq \Pr(\mathbf{HashCDH} - \mathbf{1}_7(\alpha\beta)) + \Pr(\mathbf{HashCDH} - \mathbf{0}_6(\alpha))\\ &\quad + \frac{q_s^2}{2^l} + \frac{q_h}{2^l} + \frac{q_s^2}{2^l} + \frac{q_h}{2^l} + \frac{q_h}{2^l} + \frac{(9q_s + q_h)^2}{2^l} + 1/2\\ &\leq \Pr(\mathbf{HashCDH} - \mathbf{1}_7(\alpha\beta) \cup \mathbf{HashCDH} - \mathbf{0}_6(\alpha))\\ &\quad + \frac{2q_s^2 + 3q_h + (9q_s + q_h)^2}{2^l} + 1/2\end{aligned}$$

TABLE 4. Comparison of the estimated time cost of each protocol (in ms).

Protocols	V_i	Rj	TA
[3]	$3T_e + 6T_h + 1T_{enc} = 14.4$	0	$4T_e + (3 + 0.5n)T_h + 1T_{enc} = 19.25 + 0.015n$
[2]	$1T_e + 8T_h + 1T_{enc} = 5.12$	$1T_h = 0.03$	$0.5nT_e + (5 + n)T_h + 1T_{enc} = 0.27 + 2.41n$
[4]	$7T_h + 3T_m = 60.9$	$6T_h + 3T_m = 60.87$	$10T_h + 2T_m = 40.76$
Our Protocol	$3T_e + 7T_h + 1T_{enc} = 14.61$	$1T_h = 0.03$	$3T_e + 4T_h + 1T_{enc} = 14.52$

$$\le (q_s \cdot q_h)\mathbf{Suc}^{cdh}(\mathbf{G}_7) + \frac{2q_s^2 + 3q_h + (9q_s + q_h)^2}{2^l} + 1/2$$

$$Ad_P^{se}(A)$$

$$= 2\Pr(Suc_0) - 1$$

$$\le 2(q_s \cdot q_h)\mathbf{Suc}^{cdh}(\mathbf{G}_7) + \frac{2q_s^2 + 3q_h + (9q_s + q_h)^2}{2^{l-1}}.$$

Thus, if $Ad_P^{se}(A)$ is a non-negligible function, it also implies that $\mathbf{Suc}^{cdh}\mathbf{G}_7$ is non-negligible, which contradicts our CDH assumption. By contradiction, our protocol cannot be broken by a polynomial-time adversary.

VIII. COMPARISON

In this section, we compare our proposed protocol with some related protocols [2]–[4]. Table 1 compares this protocol and its related protocols in terms of protecting user anonymity, resisting internal attacks, and resisting stealing smart card attacks. As shown, our proposed protocol and that of [4] exhibit better security.

Before calculating the cost, we must determine the time of different operations involved in the compared protocols. According to the description of each protocol, the complex operations involved in the protocol include modular exponentiation, hash, symmetric encryption/decryption, and multiplication of points on the elliptic curve. In addition, the cost of simple operations such as XOR and connection operations can often be neglected; therefore, they are not included in the calculation cost of the protocol. Our experiment uses a smartphone (iPhone 6s) as the test platform for protocol performance, and the specific parameters are as follows: system iOS 10.11, CPU Apple A9 + M9 coprocessor + up to 2.1 GHz, RAM 2GB. The complex operations involved in this experiment are the results of averaging 1000 executions, as shown in Table 5 below.

TABLE 5. Comparison of the storage cost of each protocol (in bits).

Protocols	V_i	R_j	TA
[3]	3584	256	$128 + 256n_1 + 256n_2$
[2]	3584	256	$128 + 256n_1 + 256n_2$
[4]	1664	384	$256n_1 + 256n_2$
Our Protocol	3584	256	$256n_1$

From the table, we can conclude that with the increasing number of registered users, the time for landing and certification increases as well. Because our protocol optimizes the steps of vehicle authentication, we can reduce the time cost.

Next, we compare the storage costs; however, we first provide some assumptions: the length of the ID is 256 bits; the length of the timestamp is 128 bits; the length of the hash output is 256 bits; the length of the modular power is 1024 bits; and the length of the elliptical curve (secp256r1, P and Q are 256-bits) point multiplication output is 512 bits. Table 4 shows the results in which the number of RSUs is n_1 and the number of registered vehicle users is n_2.

IX. CONCLUSION

We herein analyzed Ying *et al.*'s anonymous and lightweight authentication protocol. They claimed that their protocol could resist various attacks and was superior to other protocols. However, we found that their protocol could not achieve the claimed goals as it suffered from an offline identity guessing attack and was time consuming during the authentication phase. To enhance the security and reduce the time required for authentication, we later proposed a patch on Ying *et al.*'s protocol. The analysis proved that the patched protocol was more secure and efficient than the original protocol.

REFERENCES

[1] K. H. W. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous, Trustworthy Comput.*, vol. 1, Jun. 2006, pp. 8–15.

[2] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10626–10636, Dec. 2017.

[3] B. Ying and A. Nayak, "Efficient authentication protocol for secure vehicular communications," in *Proc. IEEE 79th Veh. Technol. Conf. (VTC Spring)*, May 2014, pp. 1–5.

[4] H. Zhong, B. Huang, J. Cui, Y. Xu, and L. Liu, "Conditional privacy-preserving authentication using registration list in vehicular ad hoc networks," *IEEE Access*, vol. 6, pp. 2241–2250, 2018.

[5] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proc. Workshop Hot Topics Netw. (HotNets-IV)*, Annapolis, MD, USA, 2005, pp. 1–6.

[6] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. IEEE INFOCOM 24th Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 3, Mar. 2005, pp. 1917–1928.

[7] L. Lazos, R. Poovendran, and S. Čapkun, "ROPE: Robust position estimation in wireless sensor networks," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.* Piscataway, NJ, USA: IEEE Press, 2005, p. 43.

[8] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, and V. Snášel, "α-fraction first strategy for hierarchical model in wireless sensor networks," *J. Internet Technol.*, vol. 19, no. 6, pp. 1717–1726, 2018.

[9] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient VANET authentication," *J. Commun. Netw.*, vol. 11, no. 6, pp. 574–588, Dec. 2009.

[10] X. Lin, X. Sun, X. Wang, C. Zhang, P. H. Ho, and X. Shen, "TSVC: Timed efficient and secure vehicular communications with privacy preserving," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4987–4998, Dec. 2008.

[11] B. Ying, D. Makrakis, and H. T. Mouftah, "Privacy preserving broadcast message authentication protocol for VANETs," *J. Netw. Comput. Appl.*, vol. 36, no. 5, pp. 1352–1364, 2013.

[12] H. Xiong, H. Zhang, and J. Sun, "Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2018.2865221.

[13] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM 27th Conf. Comput. Commun.*, Apr. 2008, pp. 246–250.

[14] C. Zhang, P.-H. Ho, and J. Tapolcai, "On batch verification with group testing for vehicular communications," *Wireless Netw.*, vol. 17, no. 8, p. 1851, 2011.

[15] M. Eltoweissy, S. Olariu, and M. Younis, "Towards autonomous vehicular clouds," in *Proc. Int. Conf. Ad Hoc Netw.* Berlin, Germany: Springer, 2010, pp. 1–16.

[16] D. Bernstein, N. Vidovic, and S. Modi, "A cloud PAAS for high scale, function, and velocity mobile applications-with reference application as the fully connected car," in *Proc. IEEE 5th Int. Conf. Syst. Netw. Commun. (ICSNC)*, Aug. 2010, pp. 117–123.

[17] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, "Rethinking vehicular communications: Merging VANET with cloud computing," in *Proc. IEEE 4th Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2012, pp. 606–609.

[18] K.-A. Shim, "CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 4, pp. 1874–1883, May 2012.

[19] Z. Jianhong, X. Min, and L. Liying, "On the security of a secure batch verification with group testing for VANET," *Int. J. Netw. Secur.*, vol. 16, no. 5, pp. 351–358, 2014.

[20] C.-C. Lee and Y.-M. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Netw.*, vol. 19, no. 6, pp. 1441–1449, 2013.

[21] J. K. Liu, T. H. Yuen, M. H. Au, and W. Susilo, "Improvements on an authentication scheme for vehicular sensor networks," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2559–2564, 2014.

[22] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Math. Comput. Model.*, vol. 55, no. 1, pp. 214–222, 2012.

[23] D. Zhao, H. Peng, L. Li, and Y. Yang, "A secure and effective anonymous authentication scheme for roaming service in global mobility networks," *Wireless Pers. Commun.*, vol. 78, no. 1, pp. 247–269, 2014.

[24] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. INFOCOM IEEE 27th Conf. Comput. Commun.*, Apr. 2008, pp. 1229–1237.

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

[26] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, "On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags," *J. Supercomput.*, vol. 74, no. 1, pp. 65–70, 2018.

[27] C.-M. Chen, B. Xiang, K.-H. Wang, K.-H. Yeh, and T.-Y. Wu, "A robust mutual authentication with a key agreement scheme for session initiation protocol," *Appl. Sci.*, vol. 8, no. 10, p. 1789, 2018.

[28] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT*, B. Preneel, Ed. Berlin, Germany: Springer, 2000, pp. 139–155.

CHIEN-MING CHEN received the Ph.D. degree from the National Tsing Hua University, Taiwan. He is currently an Associate Professor with the College of Computer Science and Technology, Shandong University of Science and Technology, Shandong, China. His current research interests include network security, mobile internet, wireless sensor network, and cryptography.

BIN XIANG is currently pursuing the M.S. degree with the Shenzhen Graduate School, Harbin Institute of Technology, China. His current research interests include security protocol and network security.

YINING LIU received the B.S. degree in applied mathematics from Information Engineering University, Zhengzhou, China, in 1995, the M.E. degree in computer software and theory from the Huazhong University of Science and Technology, Wuhan, China, in 2003, and the Ph.D. degree in mathematics from Hubei University, Wuhan, in 2007. He is currently a Professor with the School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include information security protocol and data privacy.

KING-HANG WANG received the B.Eng. degree from The Chinese University of Hong Kong and Ph.D. degree from National Tsing Hua University. He was with the Hong Kong Institute of Technology, in 2010, as a Lecturer. He joined The Hong Kong University of Science and Technology, in 2015. His research focus is cryptography, mobile security, and provable authentication.