An Efficient Certificateless Authenticated Key Exchange Protocol Resistant to Ephemeral Key Leakage Attack for V2V Communication in IoV

Lei Meng, Haitao Xu, *Member, IEEE*, Hu Xiong, *Member, IEEE*, Xuewang Zhang,
Xianwei Zhou and Zhu Han, *Fellow, IEEE*

***Abstract*—The emergence of the Internet of Vehicles (IoV) has enhanced the comfort and safety of driving by right of the intelligent transportation system, and communication among devices and infrastructures. However, the messages exchanged between them are mainly through wireless networks, which also makes the devices of IoV vulnerable. In an effort to cope with this problem, many authenticated key exchange protocols tailored for the IoV have been proposed. However, the existing similar protocols are either insufficiently secure or suffer from efficiency issues. Therefore, we propose a new security property named non-full key escrow. Furthermore, we put forward a non-full key escrow authenticated key exchange protocol tailored for V2V communication, which can resist the ephemeral key leakage attack. Then, we perform the security proof in the eCK model and carry out a performance analysis through a series of experiments. The results provide evidence that the proposed protocol is superior in terms of efficiency while compared with existing authenticated key exchange protocols for IoV.**

***Index Terms*—Internet of Vehicles, Authenticated key exchange, Ephemeral key leakage attack, Non-full key escrow, V2V, eCK model.**

I. Introduction

INTERNET of Vehicles (IoV) can be interpreted as a service platform to realize the network connections between vehicles to vehicles, vehicles to roadside units, vehicles to network infrastructures, vehicles to personal devices, and vehicles to sensors. With the aid of information technology, the devices in IoV can communicate with each other based on the assigned IP addresses. The IoV can be perceived as integrated by the Internet of Things (IoT) technology and intelligent transportation systems (ITS). It can provide three main functions: intelligent vehicle control, intelligent traffic management, and dynamic information services [1]. With practical applications of IoV, it optimizes traffic jams by dynamically calculating optimal routes, decreasing carbon emissions by reasonable resource scheduling, and alleviating traffic accidents by recognizing and anticipating the risks [2].

In IoV system, there are three main communication components: cross-vehicular communication, inner-vehicular communication, and vehicular mobile communication [3], [4]. Considering IoV as a heterogeneous network, it contains communication among vehicles, personal devices, sensors, roadside units, and network infrastructures, as shown in Fig. 1. Among all the communication paradigms, V2V communication can help detect road conditions and assist in parking [5]. However, since the communication of the vehicle is mainly based on the wireless networks, the attackers might capture, modify and replay the messages during the vehicles communication, and even compromise the vehicles if no security measures are taken. Therefore, security is one of the most urgent problems to handle for V2V communication in IoV. The security of V2V communication involves data security and individual privacy, which are the same as the traditional Internet [6]. Considering the unique features of IoV, it would be of the essence to study a security protocol tailored for V2V communication. For the sake of security, the authentication key exchange (AKE) protocols can conduct to negotiate the common session keys for the entities of IoV to encrypt and decrypt the messages. However, the traditional AKE protocol realizes identity authentication through public key infrastructure (PKI), which may fall into the trouble of certificate management. Consequently, the identify-Based cryptograph (IBC) based AKE protocols can eliminate the certificates [7]. However, in the conventional IBC-based AKE protocols, the private keys are handled by the private key generator (PKG), which may raise new security problems.

In addition, compared to the traditional vehicular ad-hoc networks, the IoV system possesses more computation ability and storage capacity [4]. Therefore, computation ability and storage capacity will no longer be the constraints for the AKE protocol in the IoV system. Based on these, some AKE protocols can achieve good efficiency in IoV. However, most of these protocols cannot satisfy the ephemeral key

Corresponding author: Haitao Xu (email: alex_xuht@hotmail.com).This work is partially supported by National Science Foundation Project of China (No. 61971032), and US NSF CNS-2128368, CNS-2107216 and Toyota.

L. Meng is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083 China (e-mail:anthemmong@qq.com).

H. Xu is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083 China (e-mail:alex_xuht@hotmail.com)

H. Xiong is with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Sichuan 610054 China (e-mail:xionghu.uestc@gmail.com).

X. Zhang is with the School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China (e-mail: zhangxw@cqupt.edu.cn)

X. Zhou is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083 China (e-mail:xwzhouli@sina.com)

Z. Han is with the Department of Electrical and Computer Engineering in the University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, 446-701.(e-mail: zhan2@uh.edu)

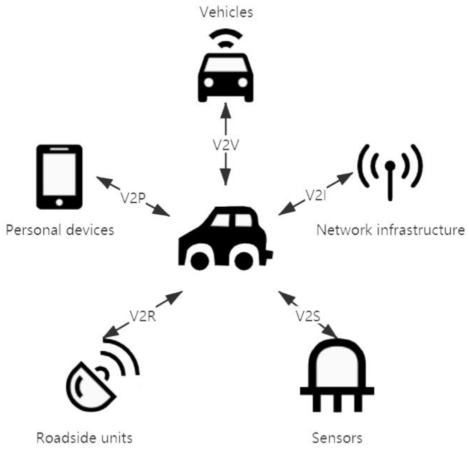


Fig. 1: Communications in IoV

leakage resilience and the non-full key escrow. The ephemeral secret leaks should not be ignored, since they are usually pre-computed and are likely to be stored in the insecure memory [8]. If the AKE protocol is full key escrow, the trust authority (TA) would know all the private keys of the entities, which results in potential security issues. In this paper, we put forward a certificateless AKE protocol resistant to ephemeral key leakage attack for V2V communication in IoV. Simultaneously, we cut down the operations of ECC-based point multiply to prompt the efficiency of the scheme. The main contributions of this paper are as follows,

1) A new security property is proposed named non-full key escrow, which can guarantee the security of the private key, even if the TA is malicious. This property can judge whether the security of the private key in the protocol is independent or not.
2) A new certificateless AKE protocol for V2V communication in IoV is proposed. It achieves non-full escrow by employing the user to compute part of the secret key and realizes ephemeral key leakage attack resistance through incorporating ephemeral secret and private key into shared information. By reducing the burdensome operations, the new protocol has better efficiency. Then, the eCK model is employed to prove its security.
3) In order to illustrate the efficiency of the proposed protocol, a series of experiments are carried out. The computation cost, communication cost, and PKG load are compared with similar protocols.

The remainder of this paper is organized as follows: we summarize the related works in Section II. After that, we introduce the preliminary knowledge of the mathematics assumptions and the eCK model in Section III. The proposed certificateless AKE protocol with ephemeral key leakage resilience for V2V communication is proposed in Section IV. Subsequently, the security proof on the proposed AKE protocol for V2V communication in the eCK model is given in Section V. Besides, the performance of the proposed AKE protocol is given and compared with some existing AKE protocols in Section VI. Finally, section VII draw a conclusion for this paper.

II. Related Work

The AKE protocol is an essential cornerstone in the field of information security. The AKE protocol can provide a pair of session keys to build a secure communication channel for two entities. The first AKE protocol was proposed by Hellman [9]. Shamir proposed an IBC based AKE protocol that can avoid the problem of certificates management under the frame of PKI [7]. Boneh and Franklin [10] put forward a practical IBC scheme adopting Weil pairing. Subsequently, many variants based on pairing were proposed. Since the pairing operation consumes too much time, the efficiency of the AKE protocol based on pairing is low [11]. In an effort to mitigate this issue, the pairing-free AKE protocol was proposed [12]. The PKG generates the private key of entities in the identity-based protocol, so the PKG can know all the private key of entities. To avoid the malicious PKG, Sattam et al. [13] proposed a certificateless AKE protocol. In certificateless AKE protocol, only part of the private key is generated by the PKG, instead of all of it. Therefore, the PKG cannot hold the whole private keys.

Alwen et al. [14] constructed a bounded leakage cryptographic scheme that can resist leakage attacks. In their scheme, the attacker knows part of the secret key information, but as long as the number of leaked bits is less than a certain limit, the attacker still cannot calculate the secret key. Then, Dodis et al. pointed out that Alwen et al.'s scheme has shortcomings in efficiency, strong security, and leakage flexibility. [15] For this reason, he proposed a new scheme based on the standard model and overcome these shortcomings. Moriyama et al. [16] mainly focused on the compromise of the ephemeral secret key in their AKE protocol. However, their schemes only concern with the long-term private leakage instead of the ephemeral secret key [17]. Chen et al. [18]pointed out that the existing security models are insufficient to capture key leakage in the real scene. To solve this problem, he ameliorated the eCK model to allow adversaries to perform key leakage queries.

The AKE protocols have also been tailored for the vehicle ad hoc network. Raya et al. [19] provided a detailed threat analysis on vehicular ad-hoc networks and designed security architecture. A set of security protocols were proposed, and analysis of security and efficiency were given. Subsequently, In the previous scheme, the vehicle receiving the message in the vehicle ad hoc network must have the support of PKI when verifying the message, which erects the problem of increased overhead. To address this issue, Zhang et al. [20] put forward an authentication scheme based on RSU-aided to improve the efficiency of authentication. Although these two schemes' primary concern is security, their efficiency still needs to be improved. The authors in [21]–[25] intent to improve the schemes' efficiency to accommodate the computation abilities of vehicle ad hoc networks.

Recently, with the rapid development of IoV, many scholars begin to focus on the AKE protocol for IoV. Liu et al. [26] developed a dual authentication protocol for IoV in various scenarios to enhance security and privacy. Then, they proved the correctness of the protocol by the Burrows-Abadi-Needham (BAN) logic [27]. But the scheme contains too many pairing operations, and the efficiency is a handicap. Mohammad et al. [28] introduced an AKE protocol for secure communication among vehicles and employed the AVISPAs tool to analyze the security. However, it is vulnerable to vehicle impersonation, fog server impersonation, cloud server impersonation attacks, and RSU impersonation [29]. Ahmed et al. [30] designed a lightweight key exchange protocol using lightweight operations, which has better efficiency. However, it suffers from ephemeral secrets leakage attacks.

The AKE protocol for V2V communication has become a popular topic in IoV systems. Inspired by previous works, we propose an AKE protocol for V2V communication to further improve the efficiency and security in this paper.

III. Preliminaries

Before introducing the proposed AKE scheme for V2V communication, some relevant elementary knowledge on mathematics difficult problems are given in Section III-A and the eCK model is presented in Section III-B.

A. Mathematics Assumptions

Denote G as a cyclic additive group of q order over E/F_p, where p is a big prime number, and let P be a generator of G. The definition of decisional Diffie-Hellman (DDH), and Gap Diffie-Hellman (GDH) are shown as follows:

Definition 1: **Decisional Diffie-Hellman (DDH):** Assuming that there are three unknown numbers $a, b, c \in Z_p^*$ and four points P, aP, bP, cP. it is impossible for a PPT algorithm to judge if $c = ab \ mod \ q$ is held or not.

Definition 2: **Gap Diffie-Hellman (GDH):** For two unknown numbers $a, b \in Z_p^*$ and declared points P, aP, bP, it is unfeasible to calculate abP by taking advantage of DDH oracle.

B. eCK Model

The original eCK model is tailored for the traditional PKI-based AKE protocol [31]. In order to be more suitable for the id-based AKE protocol, Huang and Cao modified the eCK model. In this paper, we use the modified model proposed in [32]. We introduce some essential knowledge on it, more details can be referred to the literature.

Participant Entities: Each participant entity with a unique ID_i can be perceived as a PPT Turing machine in the eCK model. Any two of them can execute the protocol one time and generate the session keys. Considering the maximum number of sessions executed by the entities as a polynomial. Let $\prod_{A,B}^N$ be the $N-$th session established by entity ID_A and entity ID_B.

Adversary: An adversary is a malicious attacker, which also can be considering as a PPT Turing machine. The adversary is able to control the whole communication channel, and it can arbitrarily eavesdrop, intercept, reply and inject messages. The adversary can do a series of queries, showing as follows, in any sequence:

1) StaticKeyReveal(ID_i): Simulator $\mathcal{M}$ returns the long-term key of entity ID_i to the adversary.
2) PKGMasterKeyReveal: Adversary $\mathcal{A}$ can obtain the master key of PKG through this query.
3) EphemeralSecretReveal($\prod_{A,B}^N$): Adversary $\mathcal{A}$ can get the ephemeral private key of entity ID_A in $\prod_{A,B}^N$.
4) SessionKeyReveal($\prod_{A,B}^N$): If state of $\prod_{A,B}^N$ is accepted, the session key will be returned.
5) Send($\prod_{A,B}^N$, m): Adversary $\mathcal{A}$ can disguise as entity ID_B and deliver a message m to $\prod_{A,B}^N$, and the $\prod_{A,B}^N$ responds according to the rule of the protocol. If m is empty, that means the entity ID_A is the initiator. If m is not empty, it needs a further judgment that the initiator is entity ID_A or not. When the answer is positive, a decision should be made (i.e. accept or reject), otherwise, returning m and making a decision.
6) Test($\prod_{A,B}^N$): This query requires that the object must be a fresh oracle, and it can be performed once in this query. Choose a number $b \in \{0, 1\}$ fairly and randomly. If $b = 0$, the simulator delivers the session key of $\prod_{A,B}^N$ to the adversary; otherwise randomly picks a number with the same distribution as the real session key and returns it.

Definition 3: **Matching Session:** Assuming that the state of $\prod_{A,B}^N$ and $\prod_{B,A}^L$ are both accepted. We say they are matching sessions for each other if they both have the same identify set constituted by initiator and responder.

Definition 4: **Fresh Oracle:** We consider $\prod_{A,B}^N$ whose state is accepted as a fresh oracle if none of the following situations happen:

1) The adversary knows the session key of $\prod_{A,B}^N$ and its matching session $\prod_{B,A}^L$ (if it exists).
2) If $\prod_{A,B}^N$'s matching session $\prod_{B,A}^L$ exists, the long-term and ephemeral private key of the entity ID_A in $\prod_{A,B}^N$ or the long-term and ephemeral private key of entity ID_B in $\prod_{B,A}^L$ are leaked to the adversary.
3) If $\prod_{A,B}^N$ has no matching session, the long-term and ephemeral private key of entity ID_A in $\prod_{A,B}^N$ or the long-term private key of entity ID_B is revealed.

IV. Proposed AKE Scheme

In this section, the proposed AKE protocol for V2V communication is introduced. There are three entities are involved in the proposed AKE protocol. The first two entities are two vehicles, which need to build a secure channel to communicate with each other. PKG is the third entity that can be considered as a trusted authority, which involves vehicles' private key generation. The proposed AKE protocol is mainly assorted into three phases: setup, private key extraction, and key exchange, respectively. Before discussing them, for ease of presentation, some symbols utilized in this paper are given in Table I.

A. Setup

In this phase, the PKG initials some vital parameters and publishes them to the entities. The steps of this phase are performed in the following manner:

TABLE I: Symbols explanation

Symbols	Explanation
PKG	Private key generator
s	Master key of PKG
P_{pub}	Public key of PKG
H_1, H_2	Collision-resistant hash function
t_i	Valid period of ω_i
$pk_i = (X_i, R_i)$	Public key of vehicle i
$sk_i = (x_i, p_i)$	Private key of vehicle i
n_i	Ephemeral private key of vehicle i
N_i	Ephemeral public key of vehicle i
ω_i	Public information of vehicle i
T_i	Transanction information of vehicle i
K_{ij}, K_{ji}	Shared secret between vehicle i vehicle j
SK_{ij}, SK_{ji}	Session key of vehicle i and vehicle j
$\longleftarrow, \longrightarrow$	Secure communication channel
$\dashleftarrow, \dashrightarrow$	Public communication channel

1) PKG selects a k-bit prime integer q, where k is PKG's security number. The PKG produces P as a generator to generate a q-order cyclic additive group G, which is over the E/F_p,
2) PGK picks $s \in Z_p^*$ as its master private key and keeps it as a secret, then computes the public key $P_{pub} = sP$.
3) PKG defines hash functions: $H_1 : \{0,1\}^* \,||G||G||$ $\rightarrow Z_p^*, H_2 : \{0,1\}^* \,||\, \{0,1\}^* \,||\, \{0,1\}^* \,||\, \{0,1\}^* \,||\, \{0,1\}^*$ $||\, \{0,1\}^* \,||G||G|| \,\rightarrow\, \{0,1\}^k$, respectively, where k is the length of the session key.
4) PKG publishes parameters $\{k, q, E/F_p, G, P, P_{pub}, H_1, H_2\}$.

B. Private Key Extraction

As shown in Fig. 2, the vehicles pass their IDs and partial public keys to the PGK, and the PKG generates the partial private key for each vehicle. More details of this phase are listed in the following manner:

1) Vehicle i randomly chooses the $x_i \in Z_p^*$ as the partial private key of it, and computes the partial public key $X_i = x_iP$. Setting $\omega_i = (ID_i||X_i||t_i)$, where ID_i is the identity of vehicle i and t_i is the valid period of ω_i. After that, vehicle i sends the ω_i to the PKG.
2) After receiving ω_i, the PKG checks the validity of the t_i. If it is not expired, randomly picks $r_i \in Z_p^*$ and calculats $R_i = r_iP$, $h_i = H_1(ID_i||X_i||R_i)$ and $p_i = r_i + h_is(mod\ q)$. PKG delivers the R_i, p_i to the vehicle i through a secure channel.
3) While receiving partial public key R_i and partial private key p_i, then the public key and private key of vehicle i are respectively: (X_i, R_i), (x_i, p_i).

The private key of vehicles can be divided into two parts, one is computed by the vehicle, and the other is produced by the PKG. Then the PKG only knows part of the private key instead of the entire key, which means the proposed AKE protocol is not key escrowed.

C. Key Exchange

Supposed that vehicles A and B involve key exchange, and both of them possess their own public key and private key. Given that vehicle A with the identity ID_A is an initiator, and vehicle B with the identity ID_B is a responder. Fig. 3 shows the procedure of this phase:

1) Vehicle A randomly chooses element $n_A \in Z_p^*$ as its ephemeral private key, and vehicle A generates its ephemeral public key by computing $N_A = n_AP$. Then computes $T_{A_1} = p_AN_A$ and $T_{A_2} = x_AN_A$, respectively, sets $T_A = T_{A_1}||T_{A_2}$ and $\omega'_A = (ID_A||X_A||t'_A)$, where t'_A is the valid period of ω'_A. Whereafter, vehicle A sends ω'_A, R_A and T_A to vehicle B.
2) Vehicle B checks the validity of the t'_A while receiving the request. If it's valid, vehicle B randomly picks ephemeral private key $n_B \in Z_p^*$,and generates ephemeral public key by computing $N_B = n_BP$. After that, vehicle B computes $T_{B_1} = p_BN_B$ and $T_{B_2} = x_BN_B$, sets $T_B = T_{B_1}||T_{B_2}$ and $\omega'_B = (ID_B||X_B||t'_B)$. Vehicle B sends ω'_B, R_B and T_B to vehicle A.
3) After vehicle A received the message of vehicle B, vehicle A checks the validity of t'_B and whether ID_B is target identity firstly. If they are both true, vehicle A computes $K_{AB_1} = n_Ap_A(R_B + H_1(ID_B||X_B||R_B)P_{pub}) + p_AT_{B_1}$, $K_{AB_2} = x_An_AT_{B_2}$ and session key $SK_{AB} = H_2(ID_A||ID_B||\omega_A||\omega_B||T_A||T_B||K_{AB_1}||K_{AB_2})$.
4) Vehicle B computes $K_{BA_1} = n_Bp_B(R_A + H_1(ID_A||X_A||R_A)P_{pub}) + p_BT_{A_1}$, $K_{BA_2} = x_Bn_BT_{A_2}$ and session key $SK_{BA} = H_2(ID_A||ID_B||\omega_A||\omega_B||T_A||T_B||K_{AB_1}||K_{AB_2})$.

So far, vehicles A and B both possess the identical session key.

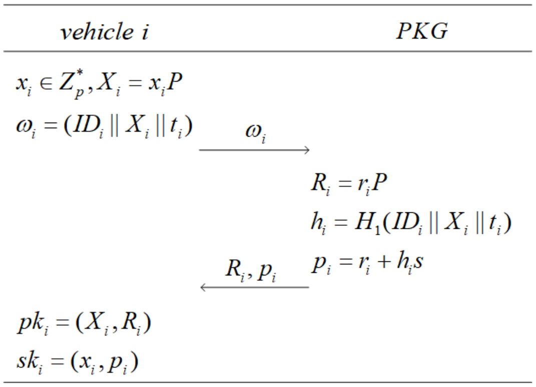

vehicle i
PKG
$x_i \in Z_p^*, X_i = x_iP$
$\omega_i = (ID_i \,||\, X_i \,||\, t_i)$
ω_i
$R_i = r_iP$
$h_i = H_1(ID_i \,||\, X_i \,||\, t_i)$
R_i, p_i
$p_i = r_i + h_is$
$pk_i = (X_i, R_i)$
$sk_i = (x_i, p_i)$

Fig. 2: Key extraction phase

D. Correctness of Proposed Protocol

To prove the consistency of the two entities' session key, we give the correctness proof in this section. Due to the only distinction are K_{AB_1}, K_{BA_1}, K_{AB_2} and K_{BA_2}, while two entities compute the session key. We just need to prove $K_{AB_1} = K_{BA_1}$, $K_{AB_2} = K_{BA_2}$, and the correctness is

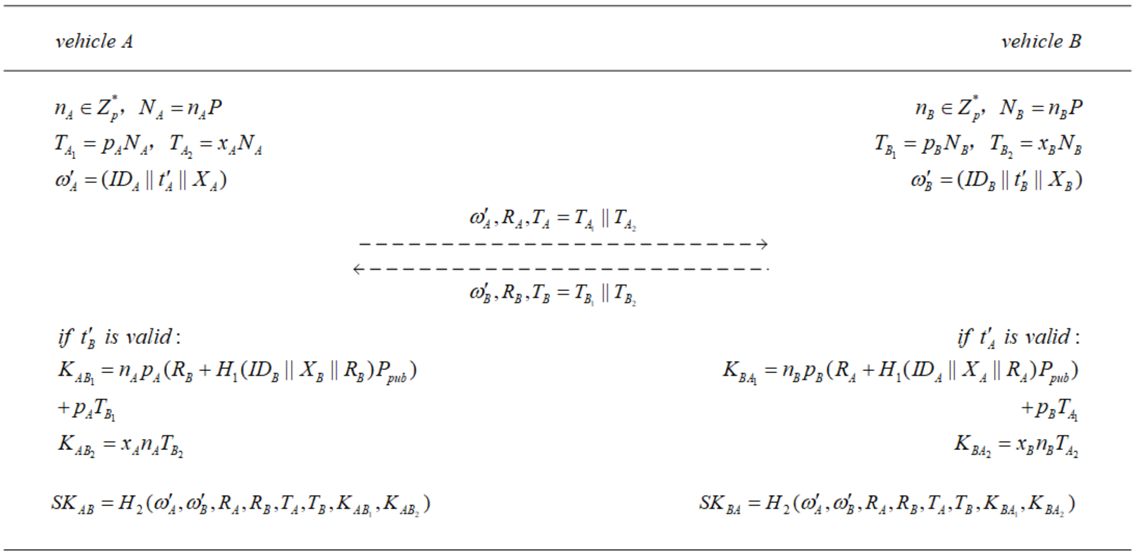

vehicle A
vehicle B
$n_A \in Z_p^*,\ N_A = n_A P$
$T_{A_1} = p_A N_A,\ T_{A_2} = x_A N_A$
$\omega'_A = (ID_A \,||\, t'_A \,||\, X_A)$
$n_B \in Z_p^*,\ N_B = n_B P$
$T_{B_1} = p_B N_B,\ T_{B_2} = x_B N_B$
$\omega'_B = (ID_B \,||\, t'_B \,||\, X_B)$
$\omega'_A, R_A, T_A = T_{A_1} \,||\, T_{A_2}$
$\omega'_B, R_B, T_B = T_{B_1} \,||\, T_{B_2}$
if t'_B is valid:
$K_{AB_1} = n_A p_A (R_B + H_1(ID_B \,||\, X_B \,||\, R_B) P_{pub})$
$+ p_A T_{B_1}$
$K_{AB_2} = x_A n_A T_{B_2}$
$SK_{AB} = H_2(\omega'_A, \omega'_B, R_A, R_B, T_A, T_B, K_{AB_1}, K_{AB_2})$
if t'_A is valid:
$K_{BA_1} = n_B p_B (R_A + H_1(ID_A \,||\, X_A \,||\, R_A) P_{pub})$
$+ p_B T_{A_1}$
$K_{BA_2} = x_B n_B T_{A_2}$
$SK_{BA} = H_2(\omega'_A, \omega'_B, R_A, R_B, T_A, T_B, K_{BA_1}, K_{BA_2})$

Fig. 3: Key exchange phase

illustrated by the following expressions:

$$\begin{aligned}
K_{AB_1} &= n_A p_A (R_B + H_1(ID_B || X_B || P_B) P_{pub}) + p_A T_{B_1} \\
&= n_A p_A (R_B + h_B P_{pub}) + p_A p_B n_B P \\
&= n_A p_A (r_B + h_B s) P + p_A p_B n_B P \\
&= n_A p_A p_B P + p_A p_B n_B P \\
&= n_B p_B p_A P + p_B p_A n_A P \\
&= n_B p_B (r_A + h_A s) P + p_B T_{A_1} \\
&= n_B p_B (R_A + H_1(ID_A || X_A || R_A) P_{pub}) + p_B T_{A_1} \\
&= K_{BA_1}.
\end{aligned}$$

For the same reason, we can prove that $K_{AB_2} = K_{BA_2}$. Now, we have proved that $K_{AB_1} = K_{BA_1}$, $K_{AB_2} = K_{BA_2}$, and so it is easy to know that $SK_{AB} = SK_{BA}$.

V. Security Analyses

In this section, we firstly discuss the security properties in Section V-A. Then we give the formal security proof of the proposed scheme in Section V-B.

A. Security Properties

The security of key exchange protocol can be judged by the security properties it satisfies. Better performance in security can be achieved when more security properties are satisfied. The main security properties are defined below.

Known-Key Security (K-KS): The authenticated key exchange protocol should be dynamic, each time the protocol performs, a unique and independent session key can be generated. The previously generated session key does not affect the security of the current session key.

Forward Security (FS): Although the long-term private key of one or more entities is leaked, it will not impact the confidentiality of the previously established session key [33]. This can fall into three situations:

1) *Partial forward security*: As long as the private keys of not all entities are leaked, the security of the previously established session key will not be impacted.
2) *Perfect forward security (PFS)*: Even if the long-term private key of all entities are leaked, the security of the previously established session keys will not be impacted [34].
3) *PKG forward security (PKG-FS)*: Despite the fact that the master key of the system held by the PKG is leaked, it will not impact the security of the previously established session key. PKG forward security is a unique security property defined for an ID-AKE protocol.

Key-Compromise Impersonation Resilience (KCIR): Assuming that the long-term private key of one entity is leaked (such as entity A), it is obvious that the adversary can disguise himself as entity A to participate in the key exchange. If the adversary cannot disguise himself as other entities, then the protocol satisfies the requirement of compromise impersonation resilience.

Unknown Key-Share (UKS): When the protocol is running, one entity (such as entity A) of the protocol establishes the session key with another entity (such as entity B), but actually, entity A establishes the session key with entity C. If the protocol can resist the occurrence of this kind of case, then the protocol satisfies the security of unknown key sharing.

Ephemeral Secrets Reveal Resistance (ESRR): The protocol can guarantee the security of the session key even if the adversary gains the ephemeral secrets of the session. Of course, the ephemeral secret leakage of this session isn't supposed to impact the security of other session keys.

Key Control (KC): The protocol satisfies key control security if the session key is determined by all the entities in the protocol instead of some of them.

Non-full Key Escrow: Escrow is that entities' private key relies on trust in the law enforcement agencies [35]. We consider a protocol as non-full key escrow if only one party of private key relies on the trusted authority. Although non-full key escrow has a strong likeness to partial key escrow, the definition of non-full escrow places more stress on no escrow in the scheme, so non-full key escrow is more appropriate as a criterion.

B. Security Proof

In the following content, the security of the proposed AKE for V2V communication is proved in the eCK model.

Definition 5: We say an AKE protocol is a secure protocol in the eCK model, while the protocol conforms to the two conditions showing in the following manner:

1) Two participant entities of the protocol can compute the same session key, respectively.
2) There isn't a PPT algorithm that can win the game with a non-negligible advantage.

Definition 6: the discrete logarithm function $DLOG : G \rightarrow Z_p^*$, when inputting $X \in G$ return $x \in Z_p^*$ such that $X = g^x$.

Theorem 1: Assume the hash function H_1, H_2 are two random oracles in the eCK model, the proposed AKE protocol is a secure protocol under the GDH assumption.

Proof: If two conditions are satisfied, the proposed AKE protocol will be a secure scheme in the eCK model. The first condition is guaranteed by the correctness of the protocol. Therefore the first condition is satisfied. Below we leverage from proof by contradiction to prove that the second condition is also met. In other words, adversary $\mathcal{A}$ can corrupt the proposed AKE protocol with non-negligible advantage, and a simulator $\mathcal{M}$ constructed by the utilization of adversary can solve the GDH problem with a non-negligible advantage.

Assume that k is the secure number, and the advantage of the adversary winning the game denotes as $Adv_A(k)$. Supposed that no more than $e(k)$ distinctive honest entities, and at most carry out n_{H_2} times H_2 query. $m(k)$ is a maximum number of the session that each entity engaged. After adversary $\mathcal{A}$ carry out the *Test* query, adversary $\mathcal{A}$ can win the game only by guessing attack, key replication attack, and forging attack. Due to the length of the session key is k byte, therefore, adversary $\mathcal{A}$ can guess the session key correctly with probability $O(1/2^k)$. H_2 outputs the same session key with probability $O(m(k)^2/2^k)$ if the two sessions are not matched. Obviously, both two of the probability is negligible.

In the remainder of this subsection, we analyze the forging attack by employing the reduction approach. Adversary $\mathcal{A}$ and simulator $\mathcal{M}$ collaborate to corrupt the protocol in the game. Simulator $\mathcal{M}$ responds to all the query proceed by adversary $\mathcal{A}$ during the whole process. If adversary $\mathcal{A}$ can win the game with a non-negligible probability $Adv_A(k)$, and then the simulator $\mathcal{M}$ can solve the GDH problem with a non-negligible probability $Adv_M(k)$ by making use of adversary $\mathcal{A}$. Given two elements $x, y \in Z_p^*$ and a GDH instance $(X = xP, Y = yP)$, with the help of DDH, the target of simulator $\mathcal{M}$ is computing $GDH(X, Y) = xyP$. At the outset, simulator $\mathcal{M}$ guesses that adversary $\mathcal{A}$ has chosen the target session $\prod_{A,B}^{N}$ with the probability larger than $1/e(k)^2 m(k)$. According to the definition of fresh oracle, two cases should be considered: (a) $\prod_{A,B}^{N}$'s matching oracle $\prod_{B,A}^{L}$ exists; (b) $\prod_{A,B}^{N}$'s matching oracle doesn't exist. In the first case, adversary $\mathcal{A}$ can be regarded as passive and it will transmit the message of ID_A and ID_B trustily. In the contrary case, adversary $\mathcal{A}$ can be considered as active, which means the message and ephemeral key of ID_A are produced by simulator $\mathcal{M}$. On the basis of the above analysis and the definition of a fresh oracle, simulator $\mathcal{M}$ must choose the strategy employed by adversary $\mathcal{A}$ from the following four situations:

S1 Passive adversary $\mathcal{A}$ doesn't know the long-term private key of entity ID_A and ephemeral private key of entity ID_B.

S2 Passive adversary $\mathcal{A}$ doesn't know the ephemeral private key of both entity ID_A and ID_B.

S3 Passive and active adversary $\mathcal{A}$ doesn't know the ephemeral private key of entity ID_A and the long-term private key of entity ID_B.

S4 Passive and active adversary $\mathcal{A}$ doesn't know the long-term private key of both entity ID_A and ID_B.

If adversary $\mathcal{A}$ can against the protocol with a non-negligible advantage by forging attack, then at least one of the situations whose probability cannot be ignored.

1) Situation S1:

Setup Phase: The simulator $\mathcal{M}$ maintains an initially empty list $list_{Setup}$, whose form likes (ID_i, R_i, ω_i). Setting the public key of PKG and long-term private key of each entity as follow:

1) Simulator $\mathcal{M}$ randomly selects $P_{pub} \in G$ as PKG's public key, and publishes parameters $param = \{k, q, E/F_p, G, P, P_{pub}, H_1, H_2\}$.
2) Simulator $\mathcal{M}$ randomly chooses $x_A, h_A \in Z_p^*$, while regarding h_A as the value of $H_1(ID_A||X_A||R_A)$. simulator $\mathcal{M}$ computes $R_A = Y - h_A P_{pub}$ and $X_A = x_A P$. Therefore, the public key of entity ID_A is (X_A, R_A), and the private key is $(x_A, \perp)$.
3) For others, simulator $\mathcal{M}$ picks $p_i, x_i, h_i \in Z_p^*$ randomly, and computes $R_i = p_i P - h_i P_{pub}$. Setting $H_1(ID_i||X_i||R_i) = h_i$, computing $X_i = x_i P$. So, the public key and private key of entity ID_i are (X_i, R_i) and (x_i, p_i), respectively.
4) For each entity ID_i, simulator $\mathcal{M}$ inserts the (ID_i, R_i, ω_i) into $list_{Setup}$, and pushes (h_i, ID_i, X_i, R_i) into the $list_{H_1}$ at the same time.

Query Phase: The simulator $\mathcal{M}$ preserves four initially empty list $list_{H_1}$, $list_{H_2}$, $list_{Reveal}$ and $list_{Send}$, which are utilized to cope with the H_1, H_2, $SessionKeyReveal$ and $Send$ queries, respectively. Simulator $\mathcal{M}$ responds to the adversary $\mathcal{A}$'s query as the following manner:

1) $H_1(ID_i, X_i, R_i)$: The simulator $\mathcal{M}$ sets the private key for entity ID_i, meanwhile, inserts (h_i, ID_i, X_i, R_i) into the $list_{H_1}$ during the setup phase. If the $list_{H_1}$ contains the (ID_i, X_i, R_i) matching with target, then return h_i. Otherwise, simulator $\mathcal{M}$ randomly picks $h_i \in Z_p^*$, and returns it to adversary $\mathcal{A}$. The simulator $\mathcal{M}$ inserts (h_i, ID_i, X_i, R_i) into $list_{H_1}$ at the same time.

2) $H_2(\omega'_i, \omega'_j, R_i, R_j, T_i, T_j, K_{ij_1}, K_{ij_2})$: Simulator $\mathcal{M}$ maintains a list $list_{H_2}$, whose initial states is empty. Each form of $list_{H_2}$ likes $(SK_{ij}, \omega'_i, \omega'_j, R_i, R_j, T_i, T_j, K_{ij_1}, K_{ij_2})$. If there is an item matching with this query, then simulator $\mathcal{M}$ sends it back to adversary $\mathcal{A}$, or simulator $\mathcal{M}$ checks if $list_{Reveal}$ contains the item. If the item is in the $list_{Reveal}$, simulator $\mathcal{M}$ sends SK_{ij} as a response, meanwhile, puts relevant item $(SK_{ij}, \omega'_i, \omega'_j, R_i, R_j, T_i, T_j, K_{ij_1}, K_{ij_2})$ into the $list_{H_2}$. If it doesn't exist in either of them, simulator $\mathcal{M}$ randomly selects $SK_{ij} \in \{0,1\}^*$ to add to $list_{Reveal}$, and return it to adversary $\mathcal{A}$.
3) $StaticKeyReveal(ID_i)$: If $i \neq A$ is not hold, simulator $\mathcal{M}$ returns (x_i, p_i) back to adversary $\mathcal{A}$. Otherwise, sends an error and exit.
4) $PKGMasterKeyReveal$: Simulator $\mathcal{M}$ sends an error and exits.
5) $EphemeralSecretReveal(\prod_{i,j}^n)$: If $\prod_{i,j}^n = \prod_{B,A}^L$, simulator $\mathcal{M}$ sends an error and exits. Otherwise, simulator $\mathcal{M}$ sends ID_i's ephemeral private key n_i back to adversary $\mathcal{A}$.
6) $Send(\prod_{i,j}^n, m)$: Simulator $\mathcal{M}$ preserves a list $list_{Send}$, whose item's form likes $(\prod_{i,j}^n, T_{i,j}^n, n_i)$, where $T_{i,j}^n$ represents a set of all message $\prod_{i,j}^n$ transmitted and received.

   If m is the first message in $T_{i,j}^n$ and $\prod_{i,j}^n = \prod_{A,B}^N$, simulator $\mathcal{M}$ obtains R_B, ID_B and X_B from the list $list_{Setup}$ and sets $n_B = \perp, \omega'_B = (ID_B||t_B'||X_B)$. Simulator $\mathcal{M}$ returns $(\omega'_B, R_B, T_{B_1} = p_B X, T_{B_2} = x_B X)$ to adversary $\mathcal{A}$ as a response, and simulator $\mathcal{M}$ inserts it into $list_{Send}$ at the same time. If $\prod_{i,j}^n \neq \prod_{A,B}^N$, simulator $\mathcal{M}$ randomly chooses $n_i \in Z_p^*$ and acquires R_i, ID_i and X_i from the list $list_{Setup}$. simulator $\mathcal{M}$ returns $(\omega'_i, R_i, p_i N_i, x_i N_i)$, meanwhile, add it to $list_{Send}$.

   If m is the second message in the set of $T_{i,j}^n$, simulator $\mathcal{M}$ sets the session is accepted.
7) $SessionKeyReveal(\prod_{i,j}^n)$: if $\prod_{i,j}^n = \prod_{A,B}^N$ or $\prod_{i,j}^n = \prod_{B,A}^L$ is hold, simulator $\mathcal{M}$ sends an error and exits. Otherwise, simulator $\mathcal{M}$ return SK_{ij} getting from $list_{Reveal}$to adversary $\mathcal{A}$.
8) $Test(\prod_{i,j}^n)$: Simulator $\mathcal{M}$ picks $SK \in \{0,1\}^k$ and sends it back to adversary $\mathcal{A}$. Otherwise, it sends an error and exits.

Analyse: If adversary $\mathcal{A}$ wins the game with a non-negligible advantage when adversary $\mathcal{A}$ chooses S1, the target session is $\prod_{A,B}^N$ and the matching session is $\prod_{B,A}^L$. Then adversary $\mathcal{A}$ must carry out the $H_2(\omega'_A, \omega'_B, R_A, R_B, T_A, T_B, K_{AB_1}, K_{AB_2})$ query, where $K_{AB_1} = n_A p_B (R_A + h_A P_{pub}) + p_B DLOG(Y)X$, $K_{AB_2} = x_A n_A T_{B_2}$. In an effort to solve the GDH problem, simulator $\mathcal{M}$ acquires items from $list_{H_2}$ and outputs $GDH(X,Y) = DLOG(Y)X = (K_{AB_1} - n_A p_B (R_A + h_A P_{pub})) p_B{}^{-1}$. The advantage of simulator $\mathcal{M}$ figuring out the GDH problem is $Adv_M(k) \geq \frac{Adv_A(k)}{4n_{H_2} e(k)^2 m(k)}$.

If $Adv_A(k)$ can't be ignored, then $Adv_M(k)$ also can't be ignored, which contradicts GDH assumption. Therefore, the advantage of adversary $\mathcal{A}$ in winning the game under situation S1 is negligible.

2) Situation S2:

Setup Phase: The simulator $\mathcal{M}$ maintains an initially empty list $list_{Setup}$, whose form likes (ID_i, R_i, ω_i). Setting the master key and public key of PKG as well as long-term private key of each entity as follow:

1) Simulator $\mathcal{M}$ randomly selects $s \in Z_p^*$ as PKG's master key and computes PKG's public key P_{pub}, and then $\mathcal{M}$ publishes parameters $param$.
2) Simulator $\mathcal{M}$ picks $p_i, x_i, h_i \in Z_p^*$ randomly, and computes $R_i = p_i P - h_i P_{pub}$. Setting $H_1(ID_i||X_i||R_i) = h_i$, computing $X_i = x_i P$. So, the public key and private key of entity ID_i are (X_i, R_i) and (x_i, p_i), respectively.
3) For each entity ID_i, simulator $\mathcal{M}$ inserts the (ID_i, R_i, ω_i) into $list_{Setup}$, and pushes (h_i, ID_i, X_i, R_i) into $list_{H_1}$ at the same time.

Query Phase: Simulator $\mathcal{M}$ preserves four initially empty list $list_{H_1}$, $list_{H_2}$, $list_{Reveal}$ and $list_{Send}$, which are exploited to cope with H_1, H_2, $SessionKeyReveal$ and $Send$ queries, respectively, where the response of $H_1(ID_i, X_i, R_i)$, $H_2(\omega'_i, \omega'_j, R_i, R_j, T_i, T_j, K_{ij_1}, K_{ij_2})$, $SessionKeyReveal(\prod_{i,j}^n)$ and $Test(\prod_{i,j}^n)$ are the same as in situation S1. Others are showing below:

1) $StaticKeyReveal(ID_i)$: Simulator $\mathcal{M}$ returns (x_i, p_i) to adversary $\mathcal{A}$.
2) $PKGMasterKeyReveal$: Simulator $\mathcal{M}$ sends master key s back to adversary $\mathcal{A}$.
3) $EphemeralSecretReveal(\prod_{i,j}^n)$: if $\prod_{i,j}^n = \prod_{A,B}^N$ or $\prod_{i,j}^n = \prod_{B,A}^L$ is hold, simulator $\mathcal{M}$ sends an error and exits. Otherwise, simulator $\mathcal{M}$ delivers the ephemeral private key n_i to adversary $\mathcal{A}$.
4) $Send(\prod_{i,j}^n, m)$: Simulator $\mathcal{M}$ preserves a list $list_{Send}$, whose item's form likes $(\prod_{i,j}^n, T_{i,j}^n, n_i)$, where $T_{i,j}^n$ represents a set of all message $\prod_{i,j}^n$ transmitted and received.

   If m is the first message in $T_{i,j}^n$ and $\prod_{i,j}^n = \prod_{A,B}^N$, simulator $\mathcal{M}$ obtains R_A, ID_A and X_A from the list $list_{Setup}$, and simulator $\mathcal{M}$ sets $n_A = \perp$, $\omega'_A = (ID_A||t_A'||X_A)$. $\mathcal{M}$ returns $(\omega'_A, R_A, T_{A_1} = p_A X, T_{A_2} = x_A X)$ to adversary $\mathcal{A}$ as response, and $\mathcal{M}$ inserts it into $list_{Send}$ at the same time. If $\prod_{i,j}^n = \prod_{B,A}^L$, simulator $\mathcal{M}$ acquires R_B, ID_B and X_B from the list $list_{Setup}$, and sets $n_B = \perp$, $\omega'_B = (ID_B||t_B'||X_B)$. simulator $\mathcal{M}$ returns $(\omega'_B, R_B, T_{B_1} = p_B Y, T_{B_2} = x_B Y)$ to adversary $\mathcal{A}$ as response and inserts it into $list_{Send}$ at the same time. Otherwise, simulator $\mathcal{M}$ randomly chooses $n_i \in Z_p^*$, and gains R_i, ID_i and X_i from the list $list_{Setup}$. simulator $\mathcal{M}$ returns $(\omega'_i, R_i, p_i N_i, x_i N_i)$, meanwhile, add it to $list_{Send}$.

   If m is the second message in the set of $T_{i,j}^n$, simulator $\mathcal{M}$ sets the session is accepted.

Analyze: If adversary $\mathcal{A}$ wins the game with a non-negligible advantage, when the adversary $\mathcal{A}$ chooses situation S2, the target session is $\prod_{A,B}^{N}$ and the matching session is $\prod_{B,A}^{L}$. Then adversary $\mathcal{A}$ must carry out the $H_2(\omega'_A, \omega'_B, R_A, R_B, T_A, T_B, K_{AB_1}, K_{AB_2})$ query, where $K_{AB_1} = Xp_Ap_B + Yp_Ap_B$, $K_{AB_2} = x_An_Ax_BY$. In an effort to solve the GDH problem, simulator $\mathcal{M}$ acquires items from $list_{H_2}$ and outputs $GDH(X,Y) = DLOG(Y)X = K_{AB_2} - {x_A}^{-1}{x_B}^{-1}$. The advantage of simulator $\mathcal{M}$ figuring out the GDH problem is $Adv_M(k) \geq \frac{Adv_A(k)}{4n_{H_2}e(k)^2m(k)}$.

If $Adv_A(k)$ can't be ignored, then $Adv_M(k)$ also can't be ignored, which contradicts GDH assumption. Therefore, the advantage of adversary $\mathcal{A}$ in winning the game under situation S2 is negligible.

3) Situation S3:

Setup Phase: The simulator $\mathcal{M}$ maintains a list $list_{Setup}$, whose form likes (ID_i, R_i, ω_i). Setting the public key of PKG and long-term private key of each entity as follow:

1) Simulator $\mathcal{M}$ randomly selects $P_{pub} \in G$ as PKG's public key, and publishes parameters $param$.
2) As for entity ID_B, simulator $\mathcal{M}$ randomly chooses $x_B, h_B \in Z_p^*$ while regarding h_B as value of $H_1(ID_B||X_B||R_B)$. simulator $\mathcal{M}$ computes $R_B = Y - h_BP_{pub}$ and $X_B = x_BP$. Therefore, the public key of entity ID_B is (X_B, R_B), and the private key is $(x_B, \perp)$.
3) Considering other entity ID_i, simulator $\mathcal{M}$ picks $p_i, x_i, h_i \in Z_p^*$ randomly, and simulator $\mathcal{M}$ computes $R_i = p_iP - h_iP_{pub}$. Setting $H_1(ID_i||X_i||R_i) = h_i$, computing $X_i = x_iP$. So, the public key and private key of entity ID_i are (X_i, R_i) and (x_i, p_i), respectively.
4) For each entity ID_i, simulator $\mathcal{M}$ inserts the (ID_i, R_i, ω_i) into $list_{Setup}$ and pushes (h_i, ID_i, X_i, R_i) into $list_{H_1}$ at the same time.

Query Phase: Simulator $\mathcal{M}$ preserves four list $list_{H_1}$, $list_{H_2}$, $list_{Reveal}$ and $list_{Send}$ to cope with the H_1, H_2, $SessionKeyReveal$ and $Send$ queries respectively. Since the adversary $\mathcal{A}$ can be active, the message and ephemeral private key might be produced by adversary $\mathcal{A}$. simulator $\mathcal{M}$ responds to the adversary $\mathcal{A}$'s query as the following manner, where the response of $H_1(ID_i, X_i, R_i)$, $H_2(\omega'_i, \omega'_j, R_i, R_j, T_i, T_j, K_{ij_1}, K_{ij_2})$, $PKGMasterKeyReveal$, $SessionKeyReveal(\prod_{i,j}^{n})$ and $Test(\prod_{i,j}^{n})$ are the same as in situation S1:

1) $StaticKeyReveal(ID_i)$: If $i \neq B$, simulator $\mathcal{M}$ returns (x_i, p_i) to adversary $\mathcal{A}$, or simulator $\mathcal{M}$ sends an error and exits.
2) $EphemeralSecretReveal(\prod_{i,j}^{n})$: If $\prod_{i,j}^{n} = \prod_{A,B}^{N}$, simulator $\mathcal{M}$ sends an error and exits. Otherwise, simulator $\mathcal{M}$ delivers ID_i's ephemeral private key n_i adversary $\mathcal{A}$.
3) $Send(\prod_{i,j}^{n}, m)$: Simulator $\mathcal{M}$ preserves a list $list_{Send}$, whose item's form likes $(\prod_{i,j}^{n}, T_{i,j}^{n}, n_i)$, where $T_{i,j}^{n}$ represents a set of all message $\prod_{i,j}^{n}$ transmitted and received.

   If m is the first message in $T_{i,j}^{n}$ and $\prod_{i,j}^{n} = \prod_{A,B}^{N}$, simulator $\mathcal{M}$ obtains R_A, ID_A and X_A from the list $list_{Setup}$ and sets $n_A = \perp$, $\omega'_A = (ID_A||{t_A}'||X_A)$. Simulator $\mathcal{M}$ returns $(\omega'_A, R_A, T_{A_1} = p_AX, T_{A_2} = x_AX)$ to adversary $\mathcal{A}$ as response, and simulator $\mathcal{M}$ inserts it into $list_{Send}$ at the same time. If $\prod_{i,j}^{n} \neq \prod_{A,B}^{n}$, simulator $\mathcal{M}$ randomly chooses $n_i \in Z_p^*$ and gains R_i, ID_i and X_i from the list $list_{Setup}$. Simulator $\mathcal{M}$ returns $(\omega'_i, R_i, p_iN_i, x_iN_i)$, meanwhile, add it to $list_{Send}$.

   If m is the second message in the set of $T_{i,j}^{n}$, simulator $\mathcal{M}$ sets the session is accepted.

Analyze: If adversary $\mathcal{A}$ wins the game with a non-negligible advantage, when adversary $\mathcal{A}$ chooses situation S3, the target session is $\prod_{A,B}^{N}$ and the matching session is $\prod_{B,A}^{L}$(if it exists). Then adversary $\mathcal{A}$ must carry out the $H_2(\omega'_A, \omega'_B, R_A, R_B, T_A, T_B, K_{AB_1}, K_{AB_2})$ query, where $K_{AB_1} = n_Bp_A(R_B + h_BP_{pub}) + p_ADLOG(Y)X$, $K_{AB_2} = x_An_AT_{B_2}$. In an effort to solve the GDH problem, simulator $\mathcal{M}$ acquires items from $list_{H_2}$ and outputs $GDH(X,Y) = DLOG(Y)X = (K_{AB_1} - n_Bp_A(R_B + h_BP_{pub})){p_A}^{-1}$. The advantage of simulator $\mathcal{M}$ figuring out the GDH problem is $Adv_M(k) \geq \frac{Adv_A(k)}{4n_{H_2}^2e(k)^2m(k)}$.

If $Adv_A(k)$ can't be ignored, then $Adv_M(k)$ also can't be ignored, which contradicts the GDH assumption. Therefore, the advantage of adversary $\mathcal{A}$ in winning the game under situation S3 is negligible.

4) Situation S4:

Setup Phase: Simulator $\mathcal{M}$ maintains a list $list_{Setup}$, whose form likes (ID_i, R_i, ω_i). Setting the public key of PKG and long-term private key of each entity as follow:

1) Simulator $\mathcal{M}$ randomly selects $P_{pub} \in G$ as PKG's public key, and publishes parameters $param$.
2) About entity ID_A, simulator $\mathcal{M}$ randomly chooses $x_A, h_A \in Z_p^*$ while regarding h_A as value of $H_1(ID_A||X_A||R_A)$. Simulator $\mathcal{M}$ computes $R_A = X - h_AP_{pub}$ and $X_A = x_AP$. Therefore, the public key of entity ID_A is (X_A, R_A), and the private key is $(x_A, \perp)$.
3) About entity ID_B, simulator $\mathcal{M}$ randomly chooses $x_B, h_B \in Z_p^*$ while regarding h_B as value of $H_1(ID_B||X_B||R_B)$. Simulator $\mathcal{M}$ computes $R_B = Y - h_BP_{pub}$ and $X_B = x_BP$. Therefore, the public key of entity ID_B is (X_B, R_B), and the private key is $(x_B, \perp)$.
4) Others, simulator $\mathcal{M}$ picks $p_i, x_i, h_i \in Z_p^*$ randomly, and computes $R_i = p_iP - h_iP_{pub}$. Setting $H_1(ID_i||X_i||R_i) = h_i$, computing $X_i = x_iP$. So, the public key and private key of the entity ID_i are (X_i, R_i) and (x_i, p_i), respectively.
5) As for each entity ID_i, simulator $\mathcal{M}$ inserts the (ID_i, R_i, ω_i) into $list_{Setup}$ and pushes (h_i, ID_i, X_i, R_i) into $list_{H_1}$ at the same time.

Query Phase: Simulator $\mathcal{M}$ preserves four list $list_{H_1}$, $list_{H_2}$, $list_{Reveal}$ and $list_{Send}$ to cope with H_1, H_2, $SessionKeyReveal$ and $Send$ queries, respectively. Since adversary $\mathcal{A}$ can be active, the message and ephemeral private key might be produced by adversary $\mathcal{A}$. Simulator $\mathcal{M}$ responds to adversary $\mathcal{A}$'s query as following manner, where the response of $H_1(ID_i, X_i, R_i)$, $H_2(\omega'_i, \omega'_j, R_i, R_j, T_i, T_j, K_{ij_1}, K_{ij_2})$,

$PKGMasterKeyReveal$, $SessionKeyReveal(\prod_{i,j}^{n})$ and $Test(\prod_{i,j}^{n})$ are the same as in situation S1:

1) $StaticKeyReveal(ID_i)$: If $i \neq A$ and $i \neq B$, simulator $\mathcal{M}$ returns (x_i, p_i) to adversary $\mathcal{A}$, or simulator $\mathcal{M}$ sends an error and exits.
2) $EphemeralSecretReveal(\prod_{i,j}^{n})$: Simulator $\mathcal{M}$ delivers ID_i's ephemeral private key n_i to adversary $\mathcal{A}$.
3) $Send(\prod_{i,j}^{n}, m)$: Simulator $\mathcal{M}$ preserves a list $list_{Send}$, whose form of item likes $(\prod_{i,j}^{n}, T_{i,j}^{n}, n_i)$, where $T_{i,j}^{n}$ represents a set of all messages $\prod_{i,j}^{n}$ transmitted and received.

   If m is the first message in $T_{i,j}^{n}$, simulator $\mathcal{M}$ randomly chooses $n_i \in Z_p^*$, and gains R_i, ID_i and X_i from list $list_{Setup}$. Simulator $\mathcal{M}$ returns $(\omega_i', R_i, p_i N_i, x_i N_i)$, meanwhile, add it to $list_{Send}$.

   If m is the second message in the set of $T_{i,j}^{n}$, simulator $\mathcal{M}$ sets the session is accepted.

Analyze: If adversary $\mathcal{A}$ wins the game with a non-negligible advantage, when adversary $\mathcal{A}$ chooses situation S4, the target session is $\prod_{A,B}^{N}$ and the matching session is $\prod_{B,A}^{L}$(if it exists). Then adversary $\mathcal{A}$ must carry out the $H_2(\omega_A', \omega_B', R_A, R_B, T_A, T_B, K_{AB_1}, K_{AB_2})$ query, where $K_{AB_1} = n_A DLOG(X)Y + n_B DLOG(Y)X$, $K_{AB_2} = x_A n_A T_{B_2}$. In an effort to solve the GDH problem, simulator $\mathcal{M}$ acquires items from $list_{H_2}$ and outputs $GDH(X,Y) = DLOG(Y)X = K_{AB_1}(n_A + n_B)^{-1}$. The advantage of simulator $\mathcal{M}$ figuring out the GDH problem is $Adv_M(k) \geq \frac{Adv_A(k)}{4n_{H_2}^2 e(k)^2 m(k)}$.

If $Adv_A(k)$ can't be ignored, then $Adv_M(k)$ also can't be ignored, which contradicts GDH assumption. Therefore, the advantage of adversary $\mathcal{A}$ in winning the game under situation S4 is negligible.

■

So far, we have proved that adversary $\mathcal{A}$ can't succeed in forging attack with non-negligible advantage. So the proposed AKE protocol for V2V communication is secure in the eCK model.

VI. Performance Comparison and Analyses

In this section, we firstly compare the security features of the proposed AKE protocol with existing AKE protocols for IoV and other IoT scenarios in section VI-A. The computation cost has been computed to demonstrate the protocol's efficiency, more precisely, we count their key extraction time and key exchange time in section VI-B. At last, we compare the PKG load on each object in section VI-C. The experiment environment is given in Table II. In our experiment, the Hash function used in each protocol is specified as SHA1, and the group order is 160 bit.

TABLE II: Experiment environment.

Name	Version/Arguments
Operation System	Ubuntu 18.04(64bit)
CPU	Intel(R) Core(TM) i7-6700
Memory	2GB
Hard Disk	40GB
Lib	PBC, Miracl, Crypto
Compiler	GCC 7.5.0

A. Comparison on Security

As shown in Table III is the comparison result in terms of security features. Obviously, the proposed AKE protocol meets all security features listing in the table. From the comparison result, except for the proposed scheme and Liu et al.'s [26] scheme in the table, the rest of the schemes do not satisfy non-full key escrow, which means their long-term keys are totally controlled by the PKG, which brings a certain degree of security risk. Since only part of the private key is generated by PKG in the proposed AKE protocol, PKG can't know the whole private key even if PKG is malicious. Besides, Liu et al.'scheme and Gupta et al.'s scheme can't resist ephemeral key leakage attack. The shared information is only incorporated with ephemeral secrets in their schemes, and adversary $\mathcal{A}$ may infer the shared key through obtained ephemeral secrets. Therefore, the protocol might be compromised while the vehicle's ephemeral key is leaked. In conclusion, our proposed AKE protocol is more secure than the other four homogeneous protocols.

TABLE III: Comparison in security.

Protocol	**SF1**	**SF2**	**SF3**	**SF4**	**SF5**	**SF6**	**SF7**
Li et.al [23]	√	√	√	√	√	√	×
Odelu et.al [36]	√	√	√	√	√	√	×
Liu et.al [26]	√	√	−	√	×	√	√
Gupta et.al [37]	√	√	√	√	×	√	×
Our protocol	√	√	√	√	√	√	√

* Note: SF1: forward security; SF2: perfect forward security; SF3: PKG forward security; SF4: key-compromise impersonation resilience; SF5: ephemeral secrets reveal resistance; SF6: key control; SF7: non-full key escrow; −: no condition to discuss.

B. Computation Cost and Communication Cost Analysis

We also make statistics for the computation, storage, and communication cost of the proposed AKE protocol and each comparison object. Table IV shows the result of the comparison, we list the computation, storage, and communication cost in the key extraction phase and the key exchange phase, where the storage cost denotes the storage space consumed by keys. According to Table V, the time of key extraction and key exchange can be estimated in each AKE scheme. Therefore, it roughly knows that our proposed AKE protocol is the best in terms of the key extraction cost and total cost. In the matter of storage and communication cost, as shown in the table, our proposed AKE protocol is not the best one, but nor the worst one. It's just a little inferior than Li et's protocol. Gupta et al., Li et al. and the proposed AKE protocol have only one round of communication, while others have more that might cost more time in reality. All in all, we can roughly know that the proposed AKE protocol has good performance whatever in time cost, communication cost, and storage overhead, although it might not be the best one.

For more precise learning about the computation consumption, we experiment to get the actual computation time in the key extraction and key exchange phase. We leverage Pairing-

TABLE IV: Comparison in computation, storage and communication cost.

	Computation cost				
Protocol	**Key extraction**	**Key exchange**	**Storage cost**	**Message size**	**Round**
Li et.al [23]	$6M + 4H \approx 2.00ms$	$12M + 6H \approx 3.90ms$	200 $bytes$	200 $bytes$	1
Odelu et.al [36]	$4PM_{G_1} + 3H \approx 4.31ms$	$2P + 3E + 5PM_{G_1} + 2PM_{G_2} + 12H \approx 7.57ms$	592 $bytes$	336 $bytes$	1.5
Liu et.al [26]	$2PM_{G_1} + 8H \approx 2.48ms$	$6P + 1E + 5PM_{G_1} + 12H \approx 10.07ms$	808 $bytes$	2220 $bytes$	2.5
Gupta et.al [37]	$2PM_{G_1} + 2H \approx 2.18ms$	$4P + 8PM_{G_1} + 3H \approx 11.27ms$	296 $bytes$	512 $bytes$	1
Our protocol	$\boldsymbol{4M + 2H \approx 1.30ms}$	$\boldsymbol{14M + 4H \approx 4.40ms}$	**240 $\boldsymbol{bytes}$**	**360 $\boldsymbol{bytes}$**	**1**

[*] Note: M: ECC-based point multiply; H: Hash; E: pairing-based exponent operation in G_2; PM_{G_1}: pairing-based group multiply operation in G_1; PM_{G_2}: pairing-based group multiply operation in G_2.

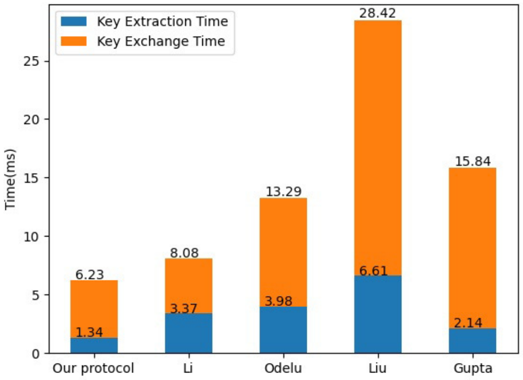
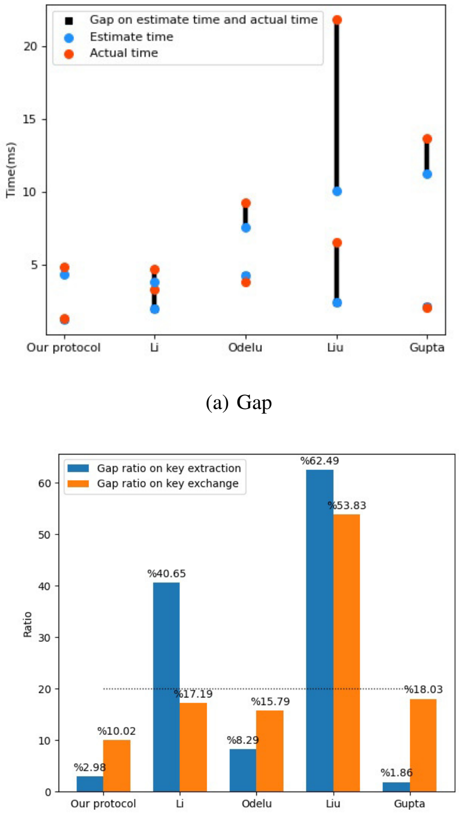


Fig. 4: Comparison in time consumption.

TABLE V: Operation time consumption(in ms).

Operation	$\boldsymbol{P}$	$\boldsymbol{E}$	$\boldsymbol{PM_{G_1}}$	$\boldsymbol{PM_{G_2}}$	$\boldsymbol{M}$	$\boldsymbol{H}$
Time comsuption	0.70	0.07	1.04	0.08	0.30	0.05



(a) Gap

(b) Gap ratio

Fig. 5: Gap between estimated time and actual time.

based Cryptography (PBC) to implement schemes based on pairings, and the implementation of ECC-based schemes is carried out by Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL). In addition, the Hash function SHA1 comes from Crypto Library. Although we employ two different Libs to implement schemes, the order of groups is both specified as 160 bit, so the error caused by Libs is negligible.[1] As shown in Fig. 4, our protocol's time consumption in the key extraction phase is the least. The total time consumption in the whole process is also the least one. The result on time consumption almost corresponds with computation cost listing in Table IV except for Liu et al.'s scheme. More details are shown in Fig. 5, the gap between estimated time and actual time in Li et al.'s scheme is also abnormal. The reason why Liu et al.'s scheme has an abnormal situation is that it contains too many data conversion and splicing operations in the whole protocol, and also includes four times data encryption and decryption operations in the key exchange phase. These operations consume too much time. The reason for Li et al.'s scheme is that its hash function includes a point multiply operation in the key extraction phase according to the definition, so it has 10 point multiply operations virtually. The gap ratio in the key extraction phase is lower in the key exchange phase except for Li et al.'s protocol and Liu et al.'s protocol. One of the factors that contribute to this situation is that the key extraction phase contains fewer data transformation and data splicing operations.

C. Analysis of PKG Load

There are a huge amount of vehicles in IoV that need to communicate, which puts a huge burden on the PKG sever. So, we statistic the PKG load in each scheme while the number of vehicles is increasing. The PKG load is mainly divided into two parts, one is the initial time consumption when PKG sets up, the other is private key extraction time. PKG sets up

[1] All the codes of experiments(includeing the comparison objects) can be found in https://github.com/AnthemMong/AKEIoV-Experiment-Codes.git

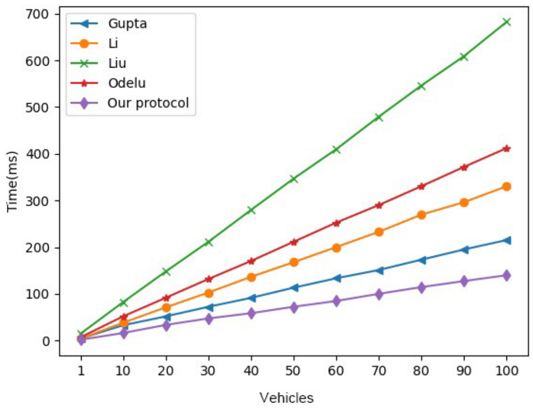


Fig. 6: Comparison in PKG time consumption.

only once in the PKG's entire life cycle, and the private key extraction of each vehicle is executed at least once. Here we assume that each vehicle extracts its private key only once. Therefore, The PKG load has a linear relationship with the number of vehicles in the system, which is consistent with what is shown in Fig. 6. Since, private key extraction of our protocol is the shortest one, with the increase of vehicles, the added PKG load is minimal too.

To sum up, the proposed AKE protocol for V2V communication in the IoV environment is more secure than the existing AKE protocols for IoV or IoT. The time consumption and PKG load are also less than them at the same time. The proposed AKE protocol's storage and communication cost are higher than some of them. In a nutshell, the proposed AKE protocol has good efficiency too.

VII. Conclusions

We propose a certificateless AKE protocol resistant to ephemeral key leakage attack for V2V communication in the IoV environment in this paper. We carry out the eCK model to prove the security of the proposed AKE protocol. The result demonstrates that our protocol meets well-known security features including resistance to ephemeral key leakage attack and non-full key escrow. Meanwhile, in terms of efficiency, we compare the time consumption, keys storage, and communication cost with existing AKE protocols. The time consumption and PKG load of our protocol are the lowest, and keys storage and communication cost are better than some of them. That means our protocol has good efficiency. As a result, our protocol is suitable for V2V communications in the IoV environment because of the high security and good efficiency.

References

[1] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 122–128, Dec. 2015.

[2] 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, Apr. 2017.

[3] M. Priyan and G. U. Devi, "A survey on internet of vehicles: applications, technologies, challenges and opportunities," *International Journal of Advanced Intelligence Paradigms*, vol. 12, no. 1-2, pp. 98–119, Dec. 2019.

[4] S. Sharma and B. Kaushik, "A survey on internet of vehicles: Applications, security issues & solutions," *Vehicular Communications*, vol. 20, pp. 1–44, Dec. 2019, doi:10.1016/j.vehcom.2019.100182.

[5] I. García-Magariño, S. Sendra, R. Lacuesta, and J. Lloret, "Security in vehicles with iot by prioritization rules, vehicle certificates, and trust management," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5927–5934, Sep. 2019.

[6] N. Sharma, N. Chauhan, and N. Chand, "Security challenges in internet of vehicles (iov) environment," in *First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. Jalandhar: IEEE, May. 2018, pp. 203–207.

[7] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology - CRYPTO 1984*. Berlin: Springer, 1984, pp. 47–53.

[8] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, Sep. 2014, doi:10.1016/j.adhoc.2014.03.009.

[9] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[10] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology - CRYPTO 2001*. Berlin: Springer, 2001, pp. 213–229.

[11] A. Karati, S. H. Islam, and G. Biswas, "A pairing-free and provably secure certificateless signature scheme," *Information Sciences*, vol. 450, pp. 378–391, Jun. 2018, doi:10.1016/j.ins.2018.03.053.

[12] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, Aug. 2010.

[13] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology - ASIACRYPT 2003*. Berlin: Springer, 2003, pp. 452–473.

[14] J. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in *Advances in Cryptology - CRYPTO 2009*. Berlin: Springer, 2009, pp. 36–54.

[15] Y. Dodis, K. Haralambiev, A. López-Alt, and D. Wichs, "Efficient public-key cryptography in the presence of key leakage," in *Advances in Cryptology - ASIACRYPT 2010*. Berlin: Springer, 2010, pp. 613–631.

[16] D. Moriyama and T. Okamoto, "Leakage resilient eck-secure key exchange protocol without random oracles," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. New York: Association for Computing Machinery, 2011, pp. 441–447.

[17] J.-D. Wu, Y.-M. Tseng, and S.-S. Huang, "Efficient leakage-resilient authenticated key agreement protocol in the continual leakage eck model," *IEEE Access*, vol. 6, pp. 17 130–17 142, Jan. 2018.

[18] R. Chen, Y. Mu, G. Yang, W. Susilo, and F. Guo, "Strong authenticated key exchange with auxiliary inputs," *Designs, Codes and Cryptography*, vol. 85, no. 1, pp. 145–173, Nov. 2017.

[19] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of computer security*, vol. 15, no. 1, pp. 39–68, Jan. 2007.

[20] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "Raise: An efficient rsu-aided message authentication scheme in vehicular communication networks," in *IEEE international conference on communications*. Beijing: IEEE, 2008, pp. 1451–1457.

[21] 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, Feb. 2017.

[22] L. Dang, J. Xu, X. Cao, H. Li, J. Chen, Y. Zhang, and X. Fu, "Efficient identity-based authenticated key agreement protocol with provable security for vehicular ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, pp. 1–16, Apr. 2018.

[23] Q. Li, C.-F. Hsu, K.-K. Raymond Choo, and D. He, "A provably secure and lightweight identity-based two-party authenticated key agreement protocol for vehicular ad hoc networks," *Security and Communication Networks*, vol. 2019, pp. 1–13, Dec. 2019, doi:10.1155/2019/7871067.

[24] L. Wu, Q. Sun, X. Wang, J. Wang, S. Yu, Y. Zou, B. Liu, and Z. Zhu, "An efficient privacy-preserving mutual authentication scheme for secure v2v communication in vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 55 050–55 063, May. 2019.

[25] 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, Mar. 2019.

[26] 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.

[27] M. Burrows, M. Abadi, and R. M. Needham, “A logic of authentication,” *Proceedings of the Royal Society of London*, vol. 426, no. 1871, pp. 233–271, Dec. 1989.

[28] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. Rodrigues, and Y. H. Park, “Akm-iov: Authenticated key management protocol in fog computing-based internet of vehicles deployment,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8804–8817, Jun. 2019.

[29] M. A. Saleem, K. Mahmood, and S. Kumari, “Comments on “akm-iov: authenticated key management protocol in fog computing-based internet of vehicles deployment”,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4671–4675, Feb. 2020.

[30] S. Ahmed, S. Kumari, M. A. Saleem, K. Agarwal, K. Mahmood, and M.-H. Yang, “Anonymous key-agreement protocol for v2g environment within social internet of vehicles,” *IEEE Access*, vol. 8, pp. 119 829–119 839, Jun. 2020.

[31] B. LaMacchia, K. Lauter, and A. Mityagin, “Stronger security of authenticated key exchange,” in *Provable Security*. Berlin: Springer, 2007, pp. 1–16.

[32] H. Huang and Z. Cao, “An id-based authenticated key exchange protocol based on bilinear diffie-hellman problem,” in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*. New York: Association for Computing Machinery, 2009, pp. 333–342.

[33] H. Sun, Q. Wen, H. Zhang, and Z. Jin, “A strongly secure identity-based authenticated key agreement protocol without pairings under the gdh assumption,” *Security and Communication Networks*, vol. 8, no. 17, pp. 3167–3179, March. 2015.

[34] M. E. S. Saeed, Q.-Y. Liu, G. Tian, B. Gao, and F. Li, “Akaiots: authenticated key agreement for internet of things,” *Wireless Networks*, vol. 25, no. 6, pp. 3081–3101, March. 2019.

[35] M. Bellare and S. Goldwasser, “Verifiable partial key escrow,” in *Proceedings of the 4th ACM Conference on Computer and Communications Security*. New York: Association for Computing Machinery, 1997, pp. 78–91.

[36] V. Odelu, A. K. Das, M. Wazid, and M. Conti, “Provably secure authenticated key agreement scheme for smart grid,” *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1900–1910, Aug. 2016.

[37] D. S. Gupta, S. H. Islam, M. S. Obaidat, P. Vijayakumar, N. Kumar, and Y. Park, “A rovably secure and lightweight identity-based two-party authenticated key agreement protocol for iiot environments,” *IEEE Systems Journal*, vol. in press, no. in press, p. in press, in press, doi:10.1109/JSYST.2020.3004551.

Lei Meng received the M.S. degree in software engineering for Chongqing University of Post and Telecommunications, China, in 2020. He is pursuiting the Ph.D degree in information and communication engineering with University of Science and Technology Beijing, China. He research interests include information security, security of internet of things.

Haitao Xu received the B.S. degree in communication engineering from Sun Yat-sen University, in 2007, the M.S. degree in communication system and signal processing from the University of Bristol, in 2009, and the Ph.D. degree from the University of Science and Technology Beijing(USTB), in 2014. He was engaged in postdoctoralstudy with the Department of Software Engineering, USTB, from 2014 to 2016. He was a Visiting Professor with the Electrical and Computer Engineering Department, University of Houston, from October 2016 to April 2017. He is currently an Associate Professor with USTB. He has published 50 articles, and one book for cyber security. His research interests include wireless communications, game theory, secure communications, cognitive radio, and mobile edge computing.

Hu Xiong received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC, Chengdu, China, in December 2009. He is currently a Professor with the School of Information and Software Engineering, UESTC. His research interests include public key cryptography and network security

Xuewang Zhang graduated from of the College of Information Engineering, Changsha Railway University, China, in 1997, and received the Master degree from Central South University of Forestry, China, in 2003. Currently, he is working as an associate professor of Chongqing University of Posts and Communications, China. His research interests include Data security and Privacy Protection, Blockchain and IoT, Bigdata and Smart Data Processing, Communications Software. He is a senior member of China Computer Federation.

Xianwei Zhou received his M.S. degree from Zhengzhou University in 1992. He obtained Ph.D. degree in Department of Transportation Engineering from Southwest Jiaotong University, P. R. China in 1999. He was engaged in post doctor study at Beijing Jiaotong University, China, from 1999 to 2000. Now, he is a professor in School of Computer and Communication Engineering, University of Science and Technology Beijing. His research interests include the security of communication networks, cloud computing and game theory.

Zhu Han (S’01–M’04-SM’09-F’14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas.His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019 and ACM distinguished Member since 2019. Dr. Han is 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: ”for contributions to game theory and distributed management of autonomous communication networks.”