A Dynamic Privacy-Preserving Key Management Scheme for Location-Based Services in VANETs

Rongxing Lu, *Member, IEEE*, Xiaodong Lin, *Member, IEEE*, Xiaohui Liang, *Student Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—In this paper, to achieve a vehicle user's privacy preservation while improving the key update efficiency of location-based services (LBSs) in vehicular ad hoc networks (VANETs), we propose a dynamic privacy-preserving key management scheme called DIKE. Specifically, in the proposed DIKE scheme, we first introduce a privacy-preserving authentication technique that not only provides the vehicle user's anonymous authentication but enables double-registration detection as well. We then present efficient LBS session key update procedures: 1) We divide the session of an LBS into several time slots so that each time slot holds a different session key; when no vehicle user departs from the service session, each joined user can use a one-way hash function to autonomously update the new session key for achieving forward secrecy. 2) We also integrate a novel dynamic threshold technique in traditional vehicle-to-vehicle (V-2-V) and vehicle-to-infrastructure (V-2-I) communications to achieve the session key's backward secrecy, i.e., when a vehicle user departs from the service session, more than a threshold number of joined users can cooperatively update the new session key. Performance evaluations via extensive simulations demonstrate the efficiency and effectiveness of the proposed DIKE scheme in terms of low key update delay and fast key update ratio.**

***Index Terms*—Dynamic key management, privacy preservation, secure location-based services (LBSs), vehicular ad hoc networks (VANETs).**

I. Introduction

THE increasing demands of improving road safety and providing attractive location-based services (LBSs) on the road have brought us a wide interest in vehicular ad hoc networks (VANETs) [1]–[3]. Due to their broad applications close to our daily lives, VANETs have recently been paid much attention from not only the government but academia and the automobile industry as well [4]. In VANETs, each vehicle is equipped with an onboard unit (OBU) communication device, which allows vehicles to not only communicate with each other, i.e., vehicle-to-vehicle (V-2-V) communication, but also communicate with roadside units (RSUs), i.e., vehicle-to-infrastructure (V-2-I) communication. Furthermore, when RSUs serve as the service gateways, each vehicle can also access the remote services on the road through these RSUs. Therefore, compared with the traditional pure infrastructure-based network, e.g., a cellular network, or noninfrastructure-based network, e.g., a mobile ad hoc network (MANET), the hybrid of V-2-V and V-2-I communications makes the VANET more promising, i.e., to provide low-cost safety- and nonsafety-related services on the road [2]. In the near future, VANETs are expected to serve as a general platform for the development of any vehicle-centered applications.

LBSs are one kind of promising and value-added applications in VANETs [5]–[9], where a service provider (SP) can make use of location information to provide various services to vehicle users in a certain area, such as finding the nearest parking lot [7], [8], or establish a location-based social network to help vehicle users who have common favorites to share some interesting information in a temporally virtual community on the road [9]. However, the flourish of LBSs in VANETs still hinges up full understanding and management of the challenges that the vehicle users are concerned with, e.g., security and privacy preservation issues [10]–[15]. Because the LBSs are value-added services, the SP only allows vehicle users registered in the service to share service contents. For example, if a vehicle user has not joined or has departed from an LBS, it cannot access the service contents. Therefore, to achieve content confidentiality in an LBS in VANETs, an efficient key management scheme should be provided. On the other hand, because a VANET is usually implemented in a civilian environment, where the locations of vehicles are tightly related to the vehicle users, if an LBS in VANETs discloses privacy information of vehicle users, i.e., identity privacy and location privacy, the LBS cannot be widely accepted by the vehicle users. Therefore, when designing an efficient key management scheme, the vehicle user's privacy preservation should be taken into consideration, which makes the design of key management more challenging.

Over the past years, many centralized and contributory key management schemes have been proposed [16]–[21]. However, due to the unique characteristics of VANETs [22], these schemes are not applicable to the LBSs in VANETs. One reason is that these schemes do not consider the user's privacy, which is required in VANETs [10]–[12]; the other one is that these schemes have also not taken account of the VANET's sparse characteristic, i.e., a limited number of RSUs are sparsely deployed in an area and vehicles may take a long time to contact

Manuscript received August 16, 2010; revised April 26, 2011; accepted July 20, 2011. Date of publication September 6, 2011; date of current version March 5, 2012. This work was supported by the Natural Sciences and Engineering Research Council of Canada. The Associate Editor for this paper was L. Yang.

R. Lu, X. Liang, and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: rxlu@bbcr.uwaterloo.ca; x27liang@bbcr.uwaterloo.ca; xshen@bbcr.uwaterloo.ca).

X. Lin is with the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON L1H 7K4, Canada (e-mail: xiaodong.lin@uoit.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2011.2164068

an RSU, which could make the key update procedure in these schemes very long. Therefore, it is critical yet challenging to design a secure and efficient key management scheme for LBSs in sparse VANET environments.

In this paper, to achieve vehicle user's privacy preservation and improve key update efficiency, we propose a dynamic privacy-preserving key management scheme called the dynamic privacy-preserving key management scheme (DIKE) for the LBSs in VANETs. With the proposed DIKE scheme, each vehicle user can be privacy-preserving authenticated before joining an LBS and can also use a pseudo-ID to conceal its real identity during a service session; meanwhile, the service session key, which is used to secure service contents' distribution, can be fast and efficiently updated for achieving forward secrecy, backward secrecy, and collusion resistance. The main contributions of this paper are threefold.

First, we introduce a privacy-preserving authentication (PPA) mechanism, which is derived from an efficient group signature [23], and can not only achieve vehicle user's privacy preservation but also restrict the possible vehicle user's double registration. Because a vehicle is not allowed double registration in the same service session, some attacks caused by double registration, e.g., the sybil attack [24], can be prevented. Note that multiple pseudonyms [10]–[12] are an efficient lightweight privacy approach in VANETs. However, since each vehicle user holds multiple unlinkable pseudonyms, we cannot prevent a compromised but unrevoked vehicle user from double registration in the same session.

Second, we present efficient service session key update procedures, particularly for sparse VANET environments. Specifically, we divide a service session into several time slots, and each time slot holds a different session key. When no vehicle departs from the service session, each joined user can use the forward-secrecy technique [25] to autonomously update the new session key to reduce the key update delay (KUD). To achieve backward secrecy, we integrate a novel dynamic threshold technique [26] in traditional V-2-V and V-2-I communications. Then, when a vehicle departs from the service session, more than a threshold number τ of joined users can also cooperatively update the new session key. As a result, the key update procedure can be accelerated. In addition, since the threshold value τ dynamically increases with the current number of departed users, the proposed DIKE scheme is more flexible and can resist possible collusion from the departed vehicle users.

Third, to validate the efficiency and effectiveness of the dynamic threshold technique integrated in the proposed DIKE scheme, we also develop a custom simulator built in Java. Simulation results show that the adopted dynamic threshold technique [26] can significantly accelerate the key update procedure in terms of low KUD and fast key update ratio (KUR).

The remainder of this paper is organized as follows: In Section II, we formalize the network model, security requirements, and identify our design goal. In Section III, we introduce some preliminaries, which serve as the basis of our proposed scheme. Then, we present the proposed DIKE scheme in Section IV, followed by security analysis and performance evaluation in Sections V and VI, respectively. We review some related works in Section VII. Finally, we draw our conclusions in Section VIII.

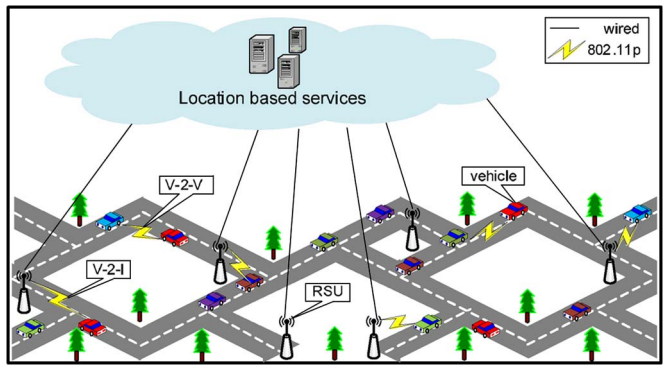


Fig. 1. Network architecture for LBSs in VANETs.

II. Network Model, Security Requirements, and Design Goal

In this section, we formalize the network model and security requirements and identify our design goals.

A. Network Model

We consider a typical LBS in VANETs, which comprises an SP, some deployed RSUs affiliated to the SP, and a large number of vehicle users $\mathcal{U}' = \{U_1, U_2, \ldots\}$ moving around the area, as shown in Fig. 1. The SP in the area can provide various services, e.g., the SP can help a vehicle user to find the nearest shopping mall to its current location, provide some local traffic information, or establish a virtual on-road community such that vehicle users who have common interests can talk with each other or broadcast messages in the virtual community. Because the vehicle users move along the road, the SP cannot directly reach the vehicles. Therefore, after being connected with the SP by wired links or any other links with high bandwidth and low delay, the affiliated RSUs can serve as the service gateways, i.e., RSUs can help the SP to broadcast and/or relay messages to vehicle users via vehicular communications.

The stationary RSUs are usually located at the road side and perform two main functions: broadcasting and relaying. The broadcasting component is responsible for broadcasting service contents that originated from the SP to the vehicle users on the road, where the service contents can either directly reach the passing-by vehicles or reach other vehicles in a multihop manner. The relaying component helps vehicle users with forwarding some requests to the SP and also helps the SP relay the responses back to the vehicle users. In some cases, it could also help the SP to preauthenticate some requests to reduce the burdens at the SP. RSU is trustable and usually equipped with not only high-storage capacity but strong computational capability as well, which causes its high cost. Then, due to the high cost, it is impractical to erect RSUs to cover the whole area, particularly at the early deployment of LBSs in VANETs. Therefore, in our network model, only a small number of RSUs are deployed at some spots.

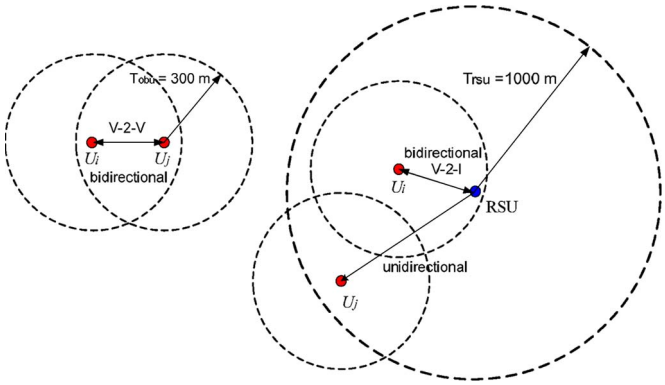


Fig. 2. Bidirectional/unidirectional communications in VANETs.

Each vehicle $U_i \in \mathcal{U}'$ is equipped with the OBU device, which allows it to communicate with other vehicles for sharing some information of common interest or communicate with the RSUs for accessing the LBSs and receiving service contents relayed by the RSUs. Different from the mobile nodes in the traditional MANET, the OBU device in VANET has no power-constrained issue and is equipped with powerful computational and communication capabilities [22]. In addition, OBU devices and RSUs can achieve the synchronized time obtained from the GPS in VANETs.

Communication model. According to [27], the medium used for communications among the neighboring vehicle users, and between vehicle users and RSUs is 5.9-GHz dedicated short range communication identified as IEEE 802.11p. Let $\mathsf{T}_{\mathsf{rsu}} = 1000$ m and $\mathsf{T}_{\mathsf{obu}} = 300$ m be the transmission ranges of the RSU and vehicle user, respectively. Then, the wireless interfaces among vehicle users are bidirectional, i.e., if U_i hears the transmission of U_j, then U_j is also able to hear U_i. However, the wireless interfaces between vehicle users and RSUs are usually unidirectional, unless the distance between them is no greater than $\mathsf{T}_{\mathsf{obu}}$, as shown in Fig. 2. Since our focus is on dynamic key management for LBSs in VANETs, we consider an efficient collision-avoidance MAC protocol employed in the lower layer [28] and ignore the interference among vehicles. Then, when V-2-V or V-2-I communication takes place, the transmission will succeed with low latency.

B. Security Requirements

Security is crucial for the distribution of service contents in dynamic VANETs. The basic security requirements that should be satisfied by the LBS include those listed here.

1) *Vehicle user authentication.* Authenticate a vehicle user when it requests joining the LBS, e.g., when it sends a request to the SP via a nearby RSU. Meanwhile, since the LBS in VANETs is usually provided in the civilian application, the privacy of vehicle users should also be protected [13]. Otherwise, the LBS in VANETs cannot be widely accepted by the public. As a result, the vehicle user's authentication should be privacy preserving.
2) *Confidentiality.* Protect service contents from passive eavesdroppers, i.e., if a vehicle user currently does not join the LBS, it cannot access the current service contents. Specifically, to achieve the confidentiality in an LBS session, a secure session key, which is used to encrypt service contents, should only be shared by all joined vehicle users. Furthermore, the session key needs to be rekeyed for achieving forward secrecy, backward secrecy, and collusion resistance in our system, where the following conditions hold.
   a) *Forward secrecy*: When a vehicle user joins an LBS session, it can access no preceding service contents issued by the LBS.
   b) *Backward secrecy*: When a vehicle user departs from an LBS session, it also cannot continually obtain future service contents from LBS.
   c) *Collusion resistance*: When several vehicle users depart from an LBS session, even though they collude, they still cannot continually obtain the ongoing LBS.

   Because the LBS in VANETs is usually provided in the civilian environment, we only protect service contents from eavesdropping by a passive adversary, who is currently not included in the LBS session. To support the aforementioned security requirements, we assume that there exists a top-level trusted authority (TA) who bootstraps the whole system, i.e., distributing the key materials to the SP and all vehicle users. Note that, in this work, we do not consider the inside adversary, i.e., a joined vehicle user does not collude with other unjoined/departed vehicle users for disclosing the service contents. Nevertheless, we acknowledge that the attacks from the inside adversary are more challenging than those from the outside adversary in VANETs, and our future work will focus on this issue.

C. Design Goal

Under the aforementioned network model and security requirements, our design goal is to develop a privacy-preserving dynamic key management scheme for LBS in VANETs. Specifically, two desirable objectives should be achieved.

1) *PPA.* For any LBS in VANETs, as a prerequisite for resisting various attacks, authentication needs to be conducted at the beginning of communications between a vehicle and the SP or among vehicles. However, in the civilian environment, a vehicle user may not be likely to expose its real identity to either the SP or other vehicle users. Therefore, our design should support the PPA in the vehicle-user-joining phase, i.e., the SP can check the validity of vehicle users without knowing their real identities. In addition, the double-registration check should also be conducted to mitigate the possible sybil attacks to the LBS [24].
2) *Fast and secure session key update considering forward secrecy, backward secrecy, and collusion resistance.* For any LBS in VANETs, a secure session key should be employed and periodically updated when vehicle users join/depart from an LBS session to achieve forward secrecy, backward secrecy, and collusion resistance. However, unlike other networks, VANETs are very sparse, which may cause a long delay when the key update procedures

are executed. Therefore, our design should also enable vehicle users to autonomously update the session key with a one-way hash function to achieve forward secrecy for the *user-joining event* and use the VANET's unique characteristic, i.e., unidirectional communication from RSUs to vehicle and the dynamic threshold technique [26], to accelerate the key update procedure and achieve backward secrecy and collusion resistance for the *user departure event*.

III. Preliminaries

In this section, we outline the pairing technique [23] and introduce a PPA scheme derived from an efficient group signature [23], which will serve as the basis of the proposed DIKE scheme.

A. Pairing Technique

Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be three multiplicative cyclic groups of the same prime order p by following the notations in [23]. Let g_1 and g_2 be two generators of groups $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and ψ be an efficiently computable bilinear map from $\mathbb{G}_2$ to $\mathbb{G}_1$ such that $\psi(g_2) = g_1$. Suppose that $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are equipped with a pairing, i.e., a nondegenerated and efficiently computable bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ such that $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$ and $e(u^a, v^b) = e(u, v)^{ab} \in \mathbb{G}_T$ for all a, $b \in \mathbb{Z}_p^*$ and any $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$. We refer to [23] for a more comprehensive description of pairing technique and complexity assumptions.

Definition 1: A bilinear parameter generator $\mathcal{G}en$ is a probabilistic algorithm that takes a security parameter k as input and outputs a 7-tuple $(p, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, where p is a k-bit prime number; $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three groups with order p; $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ are two generators such that $\psi(g_2) = g_1$; and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a nondegenerated and efficiently computable bilinear map.

B. PPA Authentication With Double-Registration Check

The PPA authentication scheme can achieve privacy preservation and local double-registration check [23]; thus, it is particularly suitable for some secure and privacy-preserving LBS in VANETs. Concretely, the PPA authentication scheme includes the following three parts: 1) system initialization; 2) PPA; and 3) verification with double-registration check.

1) System Initialization: Given the security parameter k, a TA first generates the bilinear parameter $(p, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ by running $\mathcal{G}en(k)$ and chooses two secure cryptographic hash functions H_0 and H, where $H_0 : \{0,1\}^* \to \mathbb{G}_2^2$ and $H : \{0,1\}^* \to \mathbb{Z}_p^*$. In addition, TA chooses a random number $\gamma \in \mathbb{Z}_p^*$ as the master key and sets $w = g_2^\gamma$. With these settings, TA keeps the master key γ secretly and publishes the public parameter $\mathsf{pubs} = (p, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, w, H_0, H)$.

For each vehicle user U_i's registration, TA chooses a random number $x_i \in \mathbb{Z}_p^*$ such that $\gamma + x_i \neq 0$, computes $A_i = g_1^{1/\gamma + x_i}$, and returns (x_i, A_i) as the private key of U_i.

2) PPA: Assume that vehicle user U_i wants to join an LBS with session identifier sid at time T_i, and it uses its private key (x_i, A_i) to execute five steps to anonymously authenticate itself.

1) Get the unique session identifier (sid), and compute the parameters u, $v \in \mathbb{G}_1$ as
$$(\hat{u}, \hat{v}) \leftarrow H_0(\mathsf{pubs}, \mathsf{sid}) \in {\mathbb{G}_2}^2$$
$$u = \psi(\hat{u}) \in \mathbb{G}_1, v = \psi(\hat{v}) \in \mathbb{G}_1. \quad (1)$$
2) Choose a random number $\alpha \in \mathbb{Z}_p^*$, set $\delta = x_i\alpha \in \mathbb{Z}_p^*$, and compute $T_1 = u^\alpha$ and $T_2 = A_i v^\alpha$.
3) Choose three random numbers r_α, r_x, $r_\delta \in \mathbb{Z}_p^*$, and compute $R_1 = u^{r_\alpha}$, $R_2 = e(T_2, g_2)^{r_x} \cdot e(v, w)^{-r_\alpha} \cdot e(v, g_2)^{-r_\delta}$, and $R_3 = T_1^{r_x} \cdot u^{-r_\delta}$.
4) Compute a challenge value $c \in \mathbb{Z}_p^*$ as
$$c = H(\mathsf{pubs}, \mathsf{sid}, T_i, T_1, T_2, R_1, R_2, R_3). \quad (2)$$
5) Compute $s_\alpha = r_\alpha + c\alpha$, $s_x = r_x + cx_i$, and $s_\delta = r_\delta + c\delta$, and set the authentication information $\mathsf{PPA}(T_i)$ as $(\mathsf{pubs}, \mathsf{sid}, T_i, T_1, T_2, c, s_\alpha, s_x, s_\delta)$.

3) Verification With Double-Registration Check: After receiving $\mathsf{PPA}(T_i)$ at time T_i', the SP first checks whether $T_i' - T_i \leq \Delta T$, where ΔT is the expected legal time interval for transmission delay. If it does not hold, the authentication fails. Otherwise, the SP executes three steps.

1) Compute R_1, R_2, and R_3 from $\mathsf{PPA}(T_i)$ as $R_1 = u^{s_\alpha}/T_1^c$, $R_2 = e(T_2, g_2)^{s_x} e(v, w)^{-s_\alpha} e(v, g_2)^{-s_\delta} \cdot (e(T_2, w)/e(g_1, g_2))^c$, and $R_3 = T_1^{s_x} u^{-s_\delta}$.
2) Check whether the challenge $c = H(\mathsf{pubs}, \mathsf{sid}, T_i, T_1, T_2, R_1, R_2, R_3)$ holds. If it does hold, the authentication information is accepted; otherwise, it is rejected. The detailed correctness and security analyses can be referred to [23].
3) *Double-registration check.* In [23], once user U_i is revoked, its private key A_i will be posted on a revocation list RL. Then, by checking the equation $e(T_2/A_i, \hat{u}) \stackrel{?}{=} e(T_1, \hat{v})$, U_i's revocation status can be verified. In the PPA authentication, we can use the same idea to check whether a user makes a double registration in the same session. Based on (1), we know that the parameters $\hat{u}$, $\hat{v}$, u, and v are fixed for the same sid. Then, if U_i issues two PPA authentications in the same session sid, no matter whether U_i is revoked or not, the relations $e(A_i, \hat{u}) = e(T_2, \hat{u})/e(T_1, \hat{v}) = e(T_2', \hat{u})/e(T_1', \hat{v})$ always hold for the same user U_i. As a result, based on the relation $e(T_2, \hat{u})/e(T_1, \hat{v}) = e(T_2', \hat{u})/e(T_1', \hat{v})$, the SP can check U_i's double registration without knowing A_i. Note that, although the double registration can be checked in the same session, if U_i is not revoked, i.e., its private key A_i is not in the RL, U_i's registrations in different sessions still cannot be linked, which satisfies the privacy requirement in VANETs.

The length of the PPA authentication. When adopting the elliptic curve described in [14] in PPA, one can take p to be 160-bit prime and use a group $\mathbb{G}_1$, where each element is

161 bits. Assuming $T_i \in \mathbb{Z}_p^*$, then $(T_i, T_1, T_2, c, s_\alpha, s_x, s_\delta)$ is just 1122 bits. In addition, PPA is also computational efficient, i.e., it only takes little exponentiations and bilinear pairing computations [23].

IV. Proposed DIKE Scheme

In this section, we present our DIKE for LBS in VANETs, which mainly consists of four parts, i.e., system initialization, LBS settings, vehicle user joining, and vehicle user departure. Before delving into the details of our scheme, we first provide an overview of DIKE.

A. Overview

DIKE is a special-purpose dynamic key management scheme for the LBS in sparse VANETs. Based on the privacy requirements in VANETs, DIKE first provides privacy-preserving authentication for vehicle users. At the same time, since the sparse characteristic of VANETs could make the KUD very long, DIKE also divides an LBS session into several time slots; then, if no vehicle user departs from the LBS session, each vehicle user can autonomously update the session key with a one-way hash function to achieve forward secrecy. On the other hand, to achieve backward secrecy in case of the user departure event, DIKE also adopts the dynamic threshold technique [26] in the key update procedure, where more than a threshold number τ of vehicle users can cooperatively generate the new session key after receiving the key update message. Therefore, compared to the traditional key distribution method (without the cooperatively threshold key update) in VANETs, the KUD in DIKE can be reduced. Furthermore, since the adopted threshold technique is dynamic, i.e., the threshold value τ will dynamically increase with the number of departed vehicle users, which can resist the collusion attack caused by the departed vehicles.

B. System Initialization

For a single-authority VANET under consideration, it is reasonable to assume that a TA will bootstrap the whole system. Specifically, in this phase, given the security parameter k, TA first generates $(p, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ by running $\mathcal{G}en(k)$ and chooses one secure symmetric encryption algorithm $Enc()$ and two secure cryptographic hash functions H_0 and H, where $H_0 : \{0,1\}^* \to \mathbb{G}_2^2$ and $H : \{0,1\}^* \to \mathbb{Z}_p^*$. In addition, TA also chooses a random number $\gamma \in \mathbb{Z}_p^*$ as the system master key and sets $w = g_2^\gamma$. In the end, the system parameters $\mathsf{pubs} = (p, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, w, Enc(), H_0, H)$ are published.

When an SP registers itself in the system for providing some LBS, TA uses the master key γ to compute the private key $B = g_1^{(\gamma + H(\mathrm{LBS}))^{-1}}$ and send the private key B back to the SP via a secure channel. Note that the probability that $\gamma + H(\mathrm{LBS}) = 0 \bmod p$ in the system is negligible. In addition, for each vehicle user U_i's registration, TA first chooses a random number $x_i \in \mathbb{Z}_p^*$ such that $\gamma + x_i \neq 0 \bmod p$, computes $A_i = g_1^{(\gamma + x_i)^{-1}}$, and returns (x_i, A_i) as the private key of vehicle user U_i.

C. LBS Settings

Assuming that the SP wants to establish an LBS session sid in a specific area, where sid lasts a time period t and has the estimated maximal capacity m, the following steps are performed by the SP.

- *Step 1*: The SP first chooses an initial session key $\mathbf{k} \in \mathbb{Z}_p^*$, divides the time period t into several slots $0, 1, 2, \ldots, \mathbf{t}$ [25], and sets session key $\mathbf{k}_i$ in different slots i as

$$\mathbf{k}_i = \begin{cases} \mathbf{k}, & i = 0 \\ H(\mathbf{k}_{i-1}), & 1 \leq i \leq \mathbf{t}. \end{cases} \tag{3}$$

- *Step 2*: The SP chooses two random numbers r, $a \in \mathbb{Z}_p^*$, two generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$, and a dummy user set $\mathcal{D} = \{dd_1, dd_2, \ldots, dd_{m-1}\}$, which will be used for dynamic key management for the user departure event.
- *Step 3*: The SP respectively sets the LBS session sid's master key mk, encryption key ek, and combining key ck as

$$\begin{cases} \mathsf{mk} = (g, r, a) \\ \mathsf{ek} = (m, \zeta, \eta, h^a, \{h^{a \cdot r^i}\}_{i=1}^{2m-1}, \mathcal{D}) \\ \mathsf{ck} = (m, h, \{h^{r^i}\}_{i=1}^{m-2}, \mathcal{D}) \end{cases} \tag{4}$$

  where $\zeta = g^{ar}$ and $\eta = e(g, h)^a$.
- *Step 4*: The SP periodically broadcasts the LBS session sid via the RSUs deployed in the area with the following beacon message:

$$\mathsf{bea} = \{\mathsf{LBS}, \mathsf{sid}, T_i, \mathsf{IDS}(\mathsf{sid} \| T_i)\} \tag{5}$$

  where T_i is the current timestamp, and $\mathsf{IDS}(\mathsf{sid} \| T_i)$ is a secure ID-based signature algorithm [29] signed with the private key $B = g_1^{(\gamma + H(\mathrm{LBS}))^{-1}}$ to provide origin authentication on the LBS. Concretely, $\mathsf{IDS}(\mathsf{sid} \| T_i)$ has the following form $\sigma = (\widetilde{\alpha}, \widetilde{\beta})$, where

$$\widetilde{\alpha} = H\left(\mathsf{sid} \| T_i \| e(g_1, g_2)^x\right), \widetilde{\beta} = B^{x + \widetilde{\alpha}} \tag{6}$$

  with a random number $x \in \mathbb{Z}_p^*$.

Estimation of parameter m. A challenging issue in the LBS settings phase is how to determine the parameter m in advance. In the following, we use an analytic model to give a simple estimation on m. For a specific LBS session sid, we consider that the lasting time period T_s is exponentially distributed with the density function $f(t)$, the mean $1/\mu$, and the Laplace transform $f^*(s) = (\mu/\mu + s)$. Let the user-joining event be a Poisson process and t_a be the interarrival time for user joining. Then, t_a has exponential distributions with the mean $1/\lambda$, where $(1/\lambda) \ll (1/\mu)$. Let X be the random variable of users joining sid during time period T_s. The probability $X = x$ during period $T_s = t$ can be expressed as [30]

$$\Pr[X = x | T_s = t] = \frac{(\lambda t)^x}{x!} e^{-\lambda t}. \tag{7}$$

Then, for $t \geq 0$

$$\begin{aligned}\Pr[X=x] &= \int_{t=0}^{\infty} \Pr[X=x|T_s=t]f(t)dt \\ &= \int_{t=0}^{\infty} \frac{(\lambda t)^x}{x!}e^{-\lambda t}f(t)\,dt = \left(\frac{\lambda^x}{x!}\right)\int_{t=0}^{\infty} t^x e^{-\lambda t}f(t)\,dt \\ &= \left(\frac{\lambda^x}{x!}\right)\left[(-1)^x \frac{d^x f^*(s)}{ds^x}\right]\Big|_{s=\lambda} = \frac{\mu\lambda^x}{(\mu+\lambda)^{x+1}} \end{aligned} \quad (8)$$

and the expected number of X can be computed as

$$\mathsf{E}(X) = \sum_{x=1}^{\infty} x \Pr[X=x] = \frac{\lambda}{\mu}. \quad (9)$$

Based on this estimation, the SP can choose a proper parameter $m \approx (\lambda/\mu)$ for the LBS session sid in advance.

D. Vehicle User Joining

In the vehicle-user-joining algorithm, we consider that a vehicle user U_i is moving in the area. After receiving the LBS session sid's information from the RSU, it will contact a nearby RSU to join the session sid. Assume that the vehicle's velocity is $V = 60$ km/h(≈ 16.6 m/s); then, the communication interval (CI) between V-2-I in a straight road can be roughly calculated as $\text{CI} = (2 \cdot R/V) = (2 \cdot 300/16.6) = 36.1$ s. Then, the user-joining algorithm can be executed within CI in five steps.

- *Step 1*: When U_i receives the beacon message $\mathsf{bea} = \{\mathsf{LBS}, \mathsf{sid}, T_i, \mathsf{IDS}(\mathsf{sid}\|T_i)\}$ at time T_i', U_i first checks whether $T_i' - T_i \leq \Delta T$. If it does not hold, U_i believes it is a replayed beacon and neglects it. Otherwise, U_i verifies the validity of beacon by checking the signature $\mathsf{IDS}(\mathsf{sid}\|T_i)$ with

$$\widetilde{\alpha} \stackrel{?}{=} H\left(\mathsf{sid}\|T_i\|e\left(\widetilde{\beta}, g_2^{H(\text{LBS})}\cdot w\right)\cdot e(g_1,g_2)^{-\widetilde{\alpha}}\right). \quad (10)$$

  If it holds, the beacon is accepted. The correctness is given as follows:

$$\begin{aligned} & e\left(\widetilde{\beta}, g_2^{H(\text{LBS})}\cdot w\right)\cdot e(g_1,g_2)^{-\widetilde{\alpha}} \\ &= e\left(B^{x+\widetilde{\alpha}}, g_2^{H(\text{LBS})+\gamma}\right)\cdot e(g_1,g_2)^{-\widetilde{\alpha}} \\ &= e\left(g_1^{\frac{x+\widetilde{\alpha}}{H(\text{LBS})+\gamma}}, g_2^{H(\text{LBS})+\gamma}\right)\cdot e(g_1,g_2)^{-\widetilde{\alpha}} \\ &= e(g_1,g_2)^x. \end{aligned} \quad (11)$$

- *Step 2*: Vehicle user U_i takes $(\mathsf{pubs}, \mathsf{sid})$ as input to compute the parameters $(\hat{u}, \hat{v}, u, v)$, where

$$(\hat{u},\hat{v}) \leftarrow H_0(\mathsf{pubs},\mathsf{sid}), u = \psi(\hat{u}), v = \psi(\hat{v}). \quad (12)$$

  Then, U_i picks up the current timestamp T_i; chooses a nonce $e(g_1,g_2)^y$, where $y \in \mathbb{Z}_p^*$; and uses the PPA authentication to compute $\mathsf{PPA}(T_i\|g_e^x\|g_e^y)$, where

$$\mathsf{PPA}\left(T_i\|g_e^x\|g_e^y\right) = (T_i\|g_e^x\|g_e^y, T_1, T_2, c, s_\alpha, s_x, s_\delta) \quad (13)$$

  with $g_e = e(g_1,g_2) \in \mathbb{G}_T$ and sends $\mathsf{PPA}(T_i\|g_e^x\|g_e^y)$ to the nearby RSU.
- *Step 3*: After receiving $\mathsf{PPA}(T_i\|g_e^x\|g_e^y)$, the RSU can first preauthenticate the request by checking the validity of $\mathsf{PPA}(T_i\|g_e^x\|g_e^y)$. If it is valid, the RSU forwards it to the SP. Otherwise, the RSU directly rejects the request. By this means, many potential bogus requests can be directly filtered by the RSUs to reduce the heavy burdens at the SP.
- *Step 4*: When the SP receives $\mathsf{PPA}(T_i\|g_e^x\|g_e^y)$ from the RSU, it first extracts (T_1,T_2) from $\mathsf{PPA}(T_i\|g_e^x\|g_e^y)$ and double checks (T_1,T_2) with each item in set $\mathcal{C} = \{(T_1^{(1)},T_2^{(1)}),(T_1^{(2)},T_2^{(2)}),\ldots\}$ by running Algorithm 1. When the returned value of Algorithm 1 is 1, the double registration is detected. Then, the SP rejects the registration request. Otherwise, the SP puts (T_1,T_2) in $\mathcal{C}$, chooses a pseudo-ID pd_i for U_i, and computes the corresponding private key $psk_i = g^{1/r+H(pd_i)}$. In the end, the SP puts the pseudo-ID pd_i in the joined-user set $\mathcal{U}$, computes the key $k = (g_e^y)^x = g_e^{xy}$, encrypts $msg = \mathbf{k}_j\|pd_i\|psk_i\|\mathsf{ck}$ as $C = Enc_k(msg)$, and sends C back to the vehicle user U_i via the RSU.
- *Step 5*: Upon receiving $C = Enc_k(msg)$, U_i computes the key $k = (g_e^x)^y = g_e^{xy}$, recovers msg from C, and parses it into $\mathbf{k}_j\|pd_i\|psk_i\|\mathsf{ck}$. With the LBS session key $\mathbf{k}_j$, U_i can securely communicate with other vehicle users in the LBS session. Note that, when no user departs from the LBS session sid, any user can autonomously update the session key $\mathbf{k}_{j+1}$ in time slot $j+1$ by computing $\mathbf{k}_{j+1} = H(\mathbf{k}_j)$.

Algorithm 1 Double-Registration Check

```
1: procedure DOUBLEREGISTRATIONCHECK
2:     set returnvalue = 0
3:     if there exists an item (T_1^(i), T_2^(i)) ∈ C such that the
       relation e(T_2, û)/e(T_1, v̂) = e(T_2^(i), û)/e(T_1^(i), v̂)
       does hold then
4:         set returnvalue = 1/*double registration is detected*/
5:     end if
6:     return returnvalue
7: end procedure
```

E. Vehicle User Departure

In the vehicle user departure algorithm, a vehicle user informs his leaving event to the SP, and the SP generates an update session key to prevent this user from further accessing the service. Assume that vehicle user U_l with pseudo-ID pd_l departs from the LBS session sid at time slot j. U_l first reports the departure event to a nearby RSU,[1] i.e., a signature $\mathsf{IDS}(pd_l\|\text{“}leaving\text{”})$ signed by $psk_l = g^{1/r+H(pd_l)}$ in the form of $\sigma = (\widetilde{\alpha}, \widetilde{\beta})$, where

$$\widetilde{\alpha} = H\left(pd_l\|j\|e(g,h)^x\right), \widetilde{\beta} = psk_l^{x+\widetilde{\alpha}} \quad (14)$$

[1]Note that, for any LBS in VANETs, the payment issue is also imperative for its success. When vehicle users enjoy the LBS, they should pay for their services. Therefore, to avoid the additional payment, when vehicle users depart from an LBS session, they should inform the service provider. Since our focus is on dynamic key management in LBSs, LBS payment, although important, is not discussed in this work.

with a random number $x \in \mathbb{Z}_p^*$. The RSU then forwards it to the SP. After receiving and verifying the validity of the signature $\mathsf{IDS}(pd_l \| \text{“}leaving\text{”})$ by checking

$$\widetilde{\alpha} \stackrel{?}{=} H\left(pd_l \| j \| e\left(\widetilde{\beta}, g_2^{H(LBS)} \cdot h^r\right) \cdot e(g,h)^{-\widetilde{\alpha}}\right) \tag{15}$$

where h^r is a public parameter included in the combining key ck, the SP puts the pseudo-ID pd_l in a revoked list $\mathcal{RL}$. Then, the SP executes the session key update to achieve the backward security, i.e., the departed vehicle users can no longer get the new session key. Specifically, three steps will be executed.

- *Step 1*: Let $|\mathcal{U}| = s$ and $|\mathcal{RL}|$ denote the current sizes of the sets $\mathcal{U}$ and $\mathcal{RL}$, respectively. The SP first computes a threshold $\tau = 1 + |\mathcal{RL}|$, chooses a random number $k \in \mathbb{Z}_p^*$, and computes the new session key in time slot j as $\mathbf{k}_j = H(\eta^k)$.
- *Step 2*: To ensure the operability of the dynamic key management, we require the condition that $|\mathcal{U}| - |\mathcal{RL}| \geq \tau$, i.e., $s \geq 2\tau - 1$, holds at any time in the LBS session sid. Then, under this condition, the SP encrypts η^k in the form of $C = (C_1, C_2)$, where

$$\begin{cases} C_1 = \zeta^{-k} \\ C_2 = h^{k \cdot a \prod_{x_i \in \mathcal{U}} (r + H(x_i)) \cdot \prod_{x \in \mathcal{D}_{m+\tau-s-1}} (r + H(x))} \end{cases} \tag{16}$$

  and $\mathcal{D}_{m+\tau-s-1}$ denotes the subset of $\mathcal{D}$, including the $m + \tau - s - 1$ first elements of $\mathcal{D}$. According to [26], the form of ciphertext $C = (C_1, C_2)$ allows any τ joined vehicle users with the combining key ck to efficiently and cooperatively recover the new session key $\mathbf{k}_j = H(\eta^k)$ (see Algorithm 2 for details).
- *Step 3*: The SP makes a signature $\mathsf{IDS}(msg)$ on the key update message $msg = C_1 \| C_2 \| \mathcal{U} \| \mathcal{RL}$ and broadcasts $msg \| \mathsf{IDS}(msg)$ in the area via the deployed RSUs.

Algorithm 2 Session Key Update

1: **procedure** SESSIONKEYUPDATE

2: Given the key update message msg, the combining key $\mathsf{ck} = (m, h, \{h^{r^i}\}_{i=1}^{m-2}, \mathcal{D})$, τ distinct pseudo-IDs $T = \{pd_1, pd_2, \ldots, pd_\tau\}$, and the corresponding shares $\{\sigma_i = e(g, C_2)^{1/r + H(pd_i)}\}_{i=1}^{\tau}$, compute $c_{(T,\mathcal{U})} = \prod_{x \in \mathcal{U} \cup \mathcal{D}_{m+\tau-s-1} - T} H(x)$ and a polynomial $p_{(T,\mathcal{U})}$ of degree $m-2$ to cancel a part corresponding to the $m-1$ decryption *shares* (over $m + \tau - 1$) that are not in the input. Since $p_{(T,\mathcal{U})}$ is of degree $m-2$, $h^{p_{(T,\mathcal{U})}(r)}$ is computable from ck [26], where $p_{(T,\mathcal{U})}(r) = (1/r) \cdot (\prod_{x \in \mathcal{U} \cup \mathcal{D}_{m+\tau-s-1} - T}(r + H(x)) - c_{(T,\mathcal{U})})$.

3: Given T and $\{\sigma_i = e(g, C_2)^{1/r+H(pd_i)}\}_{i=1}^{\tau}$, define the function $L_{i,l} = \sigma_i^{1/\prod_{k=1}^{i}(r+H(pd_i))} = e(g, C_2)^{(1/(r+H(pd_l))) \cdot (1/\prod_{k=1}^{i}(r+H(pd_k)))}$ for any (i, l) such that $1 \leq i < l$, and pose $L_{0,l} = \sigma_l$ for $l = 1, \ldots, l$. Then, compute sequentially $L_{i,l}$ for $i = 1, \ldots, \tau - 1$ and $l = i + 1, \ldots, \tau$ by using the induction $L_{i,l} = (L_{i-1,l}/L_{i-1,l})^{1/H(pd_l) - H(pd_i)}$, and finally output $L_\tau = L_{\tau-1,\tau} = e(g, C_2)^{1/\prod_{i=1}^{\tau}(r+H(pd_i))}$.

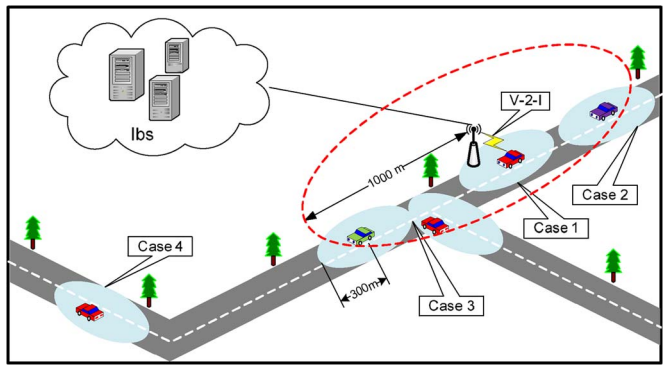


Fig. 3. Broadcasting key update message msg for the user departure event.

4: Compute η^k as

$$\begin{aligned} &\left(e\left(C_1, h^{p_{(T,\mathcal{U})}(r)}\right) \cdot L_\tau\right)^{\frac{1}{c_{(T,\mathcal{U})}}} \\ &= \left(e\left(\zeta^{-k}, h^{p_{(T,\mathcal{U})}(r)}\right) \cdot L_\tau\right)^{\frac{1}{c_{(T,\mathcal{U})}}} \\ &= \left(e\left(g^{-ark}, h^{p_{(T,\mathcal{U})}(r)}\right) \cdot e(g, C_2)^{\frac{1}{\prod_{i=1}^{\tau}(r+H(pd_i))}}\right)^{\frac{1}{c_{(T,\mathcal{U})}}} \\ &= \Bigg(e(g,h)^{-ark \cdot p_{(T,\mathcal{U})}(r)} \\ &\qquad \cdot e(g,h)^{ka \prod_{x \in \mathcal{U} \cup \mathcal{D}_{m+\tau-s-1} - T}(r+H(x))}\Bigg)^{\frac{1}{c_{(T,\mathcal{U})}}} \\ &= \left(e(g,h)^{ka \cdot c_{(T,\mathcal{U})}}\right)^{\frac{1}{c_{(T,\mathcal{U})}}} = e(g,h)^{ka} = \eta^k. \end{aligned} \tag{17}$$

5: **return** the new session key $\mathbf{k}_j = H(\eta^k)$

6: **end procedure**

When the key update message $msg \| \mathsf{IDS}(msg)$ is being broadcasted, as shown in Fig. 3, there are four cases that a vehicle can update the new session key.

Case 1: If an RSU is within the transmission range of vehicle users U_i and U_i can directly communicate with the RSU to obtain the new session key within the CI via V-2-I communication as follows.

1) U_i with pseudo-ID pd_i first chooses a random number $x \in \mathbb{Z}_p^*$, computes g^x and $\chi = psk_i^x = g^{x/r+H(pd_i)}$, makes a signature $\mathsf{IDS}(\chi)$ with the private key psk_i, and sends the request $pd_i \| \chi \| \mathsf{IDS}(\chi)$ to the RSU.
2) After receiving $pd_i \| \chi \| \mathsf{IDS}(\chi)$, the RSU first preauthenticates the request. If pd_i is not listed in the $\mathcal{RL}$ and $\mathsf{IDS}(\chi)$ is valid, the RSU forwards the request $pd_i \| \chi \| \mathsf{IDS}(\chi)$ to the SP. Otherwise, the request will be rejected.
3) When the SP receives $pd_i \| \chi \| \mathsf{IDS}(\chi)$ from the RSU, it computes

$$\chi^{r+H(pd_i)} = g^{\frac{x}{r+H(pd_i)} \cdot (r+H(pd_i))} = g^x. \tag{18}$$

   Then, it uses g^x to encrypt the new session key $\mathbf{k}_j$ as $Enc_{g^x}(pd_i \| \mathbf{k}_j)$ and sends it back to U_i via the RSU.
4) After receiving $Enc_{g^x}(pd_i \| \mathbf{k}_j)$, U_i can obtain the new session key $\mathbf{k}_j$ by recovering $Enc_{g^x}(pd_i \| \mathbf{k}_j)$ with g^x.

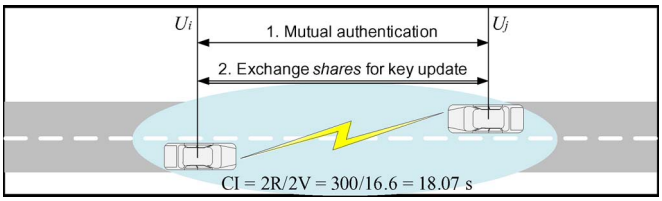


Fig. 4. Cooperative key update with V-2-V communication.

Case 2: If vehicle user U_i is driving close to an RSU and already within the transmission range of the RSU, it can receive the key update event. Then, when the RSU is within the U_i's transmission range, U_i can obtain the new session key with V-2-I communication, just the same as that in case 1.

Case 3: If vehicle user U_i is driving away from an RSU but still within the transmission range of the RSU, then U_i can still receive the key update message $msg\|\mathsf{IDS}(msg)$. After checking the validity of $\mathsf{IDS}(msg)$, U_i can use its private key $psk_i = g^{1/r+H(pd_i)}$ to compute a *share* of the new session key as follows:

$$\begin{aligned}\sigma_i &= e(psk_i, C_2) = e(g, C_2)^{\frac{1}{r+H(pd_i)}} \\ &= e(g,h)^{\frac{k\cdot a\cdot \prod_{x_i\in\mathcal{U}\cup\mathcal{D}_{m+\tau-s-1}}^{(r+H(x_i))}}{r+H(pd_i)}}. \end{aligned} \tag{19}$$

Then, U_i carries the *share* σ_i on the road. When U_i meets with other vehicle user U_j with pseudo-ID pd_j, they can possibly apply the V-2-V communication to cooperatively update the new session key. For example, when U_i meets with U_j on a straight road with the same velocity $V = 60$ km/h ($\approx$16.6 m/s), as shown in Fig. 4, the V-2-V CI can be calculated as $\text{CI} = (2 \cdot R/2 \cdot V) = (300/16.6) = 18.07$ s. Note that the CI could be longer in other situations, e.g., two vehicles stop at the intersection when the traffic light is red. Within the CI, U_i and U_j can launch three cooperative key update operations.

1) U_i chooses a random number $x \in \mathbb{Z}_p^*$, makes a signature $\mathsf{IDS}(g^x)$ with the private key $psk_i = g^{1/r+H(pd_i)}$, and sends $pd_i\|g^x\|\mathsf{IDS}(g^x)$ to U_j. Note that the key update message msg will also be forwarded to U_j if U_j has not obtained one copy.
2) After receiving $pd_i\|g^x\|\mathsf{IDS}(g^x)$, U_j checks the validity of $\mathsf{IDS}(g^x)$, and the pseudo-ID pd_i is not listed in $\mathcal{RL}$. If both of them are passed, U_j also chooses a random number $y \in \mathbb{Z}_p^*$, makes a signature $\mathsf{IDS}(g^y)$ with its private key $psk_j = g^{1/r+H(pd_j)}$, and sends $pd_j\|g^y\|\mathsf{IDS}(g^y)$ back to U_i.
3) After checking the validity of $\mathsf{IDS}(g^y)$ and pd_j, both U_i and U_j can compute the shared key g^{xy}. Then, they can communicate with each other in a secure way, i.e., all the subsequent transmitted messages are encrypted with the shared key g^{xy}. If U_j has already gotten the new session key $\mathbf{k}_j$, U_j directly sends $\mathbf{k}_j$ to U_i. Then, U_i can also gain the new session key. On the other hand, if U_j has met with other vehicle users and obtained some valid *shares*, U_i and U_j can exchange the *shares* they hold. After the *share*-exchanging procedure, if the *shares* held by any one are still less than the threshold τ, U_i and V_j will seek other vehicles on the road for key update. Otherwise, if one has reached at least τ distinct *shares*, it can use Algorithm 2 to recover the new session key $\mathbf{k}_j$ and share it with the other party. Note that, if the current time has evolved to the next time slot $j+1$, each one can compute $\mathbf{k}_{j+1} = H(\mathbf{k}_j)$ autonomously. Because of the dynamic V-2-V cooperative key update, the key update speed can be accelerated.

Case 4: Because a vehicle is beyond the transmission range of an RSU, it cannot be informed of the key update event immediately. Later, when it drives close to an RSU, it can update the session key with V-2-I communication, just like that in case 1, or, when it meets with other vehicles on the road, like that in case 3, it can also use V-2-V communication to update the session key.

Reducing the length of key update message. In the key update message $msg = C_1\|C_2\|\mathcal{U}\|\mathcal{RL}$, except the fixed-size $C_1\|C_2$, the lengths of $\mathcal{U}$ and $\mathcal{RL}$ will linearly increase with the size of $\mathcal{U}$ and $\mathcal{RL}$, respectively, which could incur high communication costs. To reduce the communication costs, we use the following policy: since set $\mathcal{U}$ is irrelevant with $\mathcal{RL}$ and when the pseudo-IDs $(pd_1, pd_2, \ldots)$ are set as having the same prefix "sid" and sequentially increasing suffix $i = 1, 2, 3, \ldots$, we can use the last $pd_{i_{\max}}$ to represent set $\mathcal{U}$. Then, the length of $\mathcal{U}\|\mathcal{RL}$ is only linear with $|\mathcal{RL}| + 1$. Because $|\mathcal{RL}| < |\mathcal{U}|$, the communication cost can be reduced. Note that the 802.11p physical layer offers different bit rates ranging from 3 to 27 Mb/s [27]; we can choose the proper bit rates for transmitting the msg so that the communication overhead is not a big issue.

Discussion on the computation cost. To further evaluate protocol complexity, we examine the computation costs at the vehicle-user-joining and departure phases in DIKE. Since the pairing computation (pa) and the point exponentiation (pe) are dominant overheads, we will only take costly online pa and pe into account. According to the execution time results for a supersingular curve of embedded degree $k = 6$ running on an Intel Pentium IV 3.0-GHz machine [14], the measured times for pa and pe are 4.5 and 0.6 ms, respectively.

When a vehicle user joins, the user first takes around $\mathsf{pe} + \mathsf{pa}$ to verify the RSU beacon message and then takes $5\mathsf{pe} + 3\mathsf{pa}$ to generate the PPA authentication message. To preauthenticate the vehicle, the RSU takes $4\mathsf{pe} + 4\mathsf{pa}$. (Some pairing operations are precomputed in advance here.) Later, the SP takes $2(n+1)\mathsf{pa}$ to run the double checking, where n is the number of registered vehicle users. In addition, the SP takes pe to generate psk for the vehicle. As a result, the computation cost in this phase is $11\mathsf{pe} + (10 + 2n)\mathsf{pa} = 51.6 + 9n$ ms. When $n = 500$, the computation cost is 4.5516 s, which is much less than the CI $\text{CI} = 36.1$ s. Because the vehicle-user-joining event does not affect other vehicles, the computation cost for others' key update is only a negligible hash operation.

When a vehicle user leaves, the user takes pe to sign the "*leaving*" event, and the SP first takes $\mathsf{pe} + \mathsf{pa}$ to verify it and then takes $3\mathsf{pe} + \mathsf{pa}$ to generate the encrypted new session key. In *Case 1*, the computation costs for the vehicle user, RSU, and SP are $3\mathsf{pe}$, $\mathsf{pe} + \mathsf{pa}$, and pe, respectively. Thus, the total cost is $5\mathsf{pe} + \mathsf{pa} = 7.5$ ms, which is negligible when compared to CI.

In *Case 2*, when a vehicle is close to the RSU, the computation cost is the same as that in case 1. In *Case 3*, both vehicles need $5\mathsf{pe} + 3\mathsf{pa} = 16.5$ ms for exchanging the *shares*. If one party holds at least τ *shares*, it takes one pairing and several exponentiation computations in $\mathbb{G}_T$ in Algorithm 2 to obtain the updated session key. In *Case 4*, the computation cost of a vehicle is contingent upon the next V-2-V and V-2-I contacts. Based on the preceding analyses, we can see that the computation cost of DIKE itself is small (in milliseconds), when compared with the delay cost (in minutes) due to the sparse VANET's architecture and vehicle mobility. Hence, the proposed DIKE scheme is applicable for VANET environments in terms of low computation complexity.

V. Security Analysis

In this section, we discuss the security properties of the proposed DIKE scheme. In particular, following the security requirements discussed earlier, our analysis will focus on how the proposed DIKE scheme can achieve the vehicle user's privacy preservation and the LBS session key's forward secrecy and backward secrecy and resist the possible collusion from the departed vehicle users. Note that, since the proposed DIKE scheme only deals with the outside adversary, other attacks launched by the inside adversary, e.g., the collusion between the joined vehicle users and the departed vehicle users, are out of the scope of this paper.

1) Proposed DIKE Scheme Can Achieve the Vehicle User's Privacy Preservation: In the vehicle-user-joining phase, since PPA authentication is employed, the real identity of vehicle user will not be disclosed. At the same time, although a vehicle user uses the same pseudo-ID during a short LBS session, the pseudo-IDs in different sessions are still different and unlinkable. Therefore, the proposed DIKE scheme can achieve the vehicle user's privacy preservation. Note that the sophisticated double-registration check executed by the SP can prevent a vehicle user from simultaneously holding more than one pseudo-ID in one session; thus, it can mitigate possible sybil attacks in VANETs.

2) Proposed DIKE Scheme Can Provide the Service Session Key's Forward Secrecy: In the proposed DIKE scheme, one LBS session is divided into several time slots $0, 1, 2, \ldots, \mathbf{t}$, and if no vehicle user departs from the session, the session key $\mathbf{k}_j$ in each time slot j is evolved from the preceding session key $\mathbf{k}_{j-1}$ with the forward security technique [25], as shown in Fig. 5. Then, because of the one-wayness of hash function $H()$, it is impossible for a vehicle user, which holds session key $\mathbf{k}_j$ at the current time slot j, to obtain any other session keys corresponding to the preceding time slots. Therefore, the forward secrecy of the LBS session key can be provided in the proposed DIKE scheme. Note that, when a vehicle user joins the LBS session in time slot j, all preceding communication contents in the same time slot j could still be available to the newly joined user since the service session key $\mathbf{k}_j$ in the same time slot is not changed. Therefore, the time slots should be carefully divided by the SP. Obviously, for a fixed session period t, the more the time slots that a session period is divided into, the higher the forward secrecy that can be achieved; however, it may incur highly frequent key update operations.

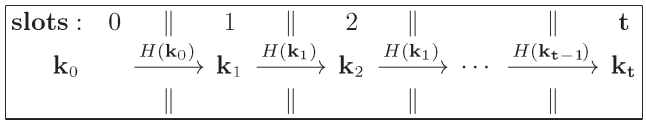


Fig. 5. Forward security key update mechanism.

3) Proposed DIKE Scheme Can Provide the Service Session Key's Backward Secrecy: In the proposed DIKE scheme, when a vehicle user departs from an LBS session, the SP will broadcast a key update message msg, which includes not only an up-to-date revoked list $\mathcal{RL}$ but a new encrypted session key independent from the preceding ones as well. Then, those joined users can still update their new session keys via either the V-2-I communication or the cooperative V-2-V communication, as described in Section IV-E. However, for those departed users, because their pseudo-IDs are listed in $\mathcal{RL}$, they cannot pass the authentication from RSUs or other vehicles. Then, they cannot obtain the new session key. As a result, the proposed DIKE scheme can achieve the service session key's backward secrecy.

4) Proposed DIKE Scheme Can Also Resist the Collusion From the Departed Vehicle Users: In the proposed DIKE scheme, each departed vehicle is still allowed to generate an efficient *share* of the new session key from the key update message msg. Then, they could collude together to combine the new session key. However, no matter how they collude, the number of efficient *shares* cannot reach the threshold τ since $\tau = |\mathcal{RL}| + 1$ dynamically increases with the current size of $\mathcal{RL}$. Therefore, when we do not consider the inside attacker, i.e., any joined vehicle user will not help the departed vehicle users, the proposed DIKE scheme is collusion resistance to the departed vehicle users.

From the preceding analysis, we can see that the proposed DIKE scheme is indeed a secure dynamic key management scheme, which can achieve the required security goals under the considered security model.

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed DIKE scheme using a custom simulator built in Java. The simulator implements the network layer and makes assumptions regarding the lower layers that the bandwidth and buffer size are always available for V-2-V and V-2-I communications. The performance metrics used in the evaluation are the following: 1) The *average KUD*, which is the average time between when a new service session key is generated by the SP and when it is successfully received by a joined vehicle user, and 2) the *average KUR*, which is defined as the ratio of the number of vehicle users who have successfully updated the new service session key to the total number of vehicle users in the same session in a time period. Both KUD and KUR can be used to examine the ability of the proposed DIKE scheme with the dynamic threshold technique [26] to accelerate the key update procedure due to a vehicle user departure event. Note that, because the vehicle-user-joining event does not affect other

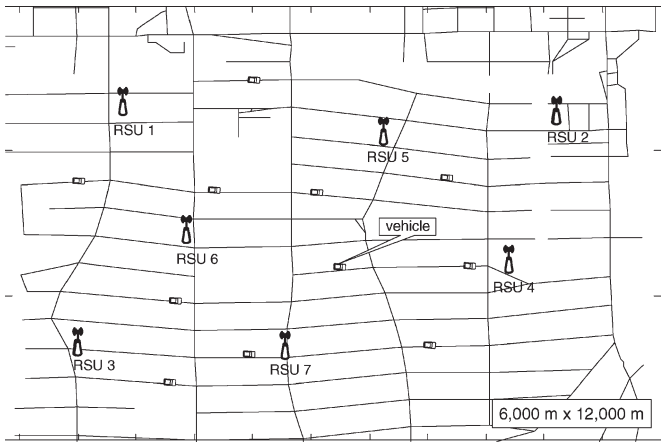


Fig. 6. Simulation area under consideration.

vehicle users' session key update, we do not consider the key update due to time evolution in our simulation.

A. Simulation Settings

In the simulations, a total of 80 *joined* vehicle users with a transmission radius of 300 m are first uniformly deployed in an interest area measuring 6000 m $\times$ 12 000 m, as shown in Fig. 6, to simulate a sparse VANET. In addition, seven locations are chosen as the candidates used for deploying the possible RSUs in the region.

Mobility model. In VANETs, due to the lesser amount of RSUs, particularly in the early stage of RSU deployment, the vehicular communications are largely contingent upon the mobility of vehicles. Since vehicles are usually driven along the roads in a city, we assume that each vehicle user follows the *shortest-path map-based movement* routing. Specifically, each vehicle first randomly chooses a destination in the area and gets there using the shortest route with the average velocity $V = 60$ km/h. After reaching the destination, with 2-min pause time, the vehicle randomly chooses a new destination and repeats the preceding steps.

We assume that an SP, together with the deployed RSUs, can provide some LBS in the area, and the user departure event takes place every 20 min, i.e., a joined vehicle user will leave the session, and a session key update will start every 20 min. To examine the outstanding performance of the proposed DIKE scheme in the key update procedure, we compare it with the traditional VANET-based key distribution manner, where any vehicle can get the updated session key from not only RSUs but other vehicles who have held the updated session key as well; however, no dynamic threshold technique is involved. The detailed parameter settings are summarized in Table I.

In the following, we run the simulations with different RSU group settings. For each setting, the simulation lasts for 400 min, and the average performance results over 20 runs are reported.

B. Simulation Results

1) Average KUD (KUD): In Fig. 7, we compare the average KUD between the proposed DIKE scheme and the traditional

TABLE I
SIMULATION SETTINGS

Parameter	Setting
Simulation	
session duration time; area	400 minutes; $6,000 \times 12,000$ m^2
Vehicle user	
number; storage; velocity; pause time	80; 50 MB; 60 ± 5 km/h; 2 min
user departure event	every 20 minutes
transmission radius; mobility model	300 m; map-based shortest path
RSU	
storage; transmission radius	10000 MB; 1000 m
groups 1, 2, 3, 4	$\{1 \cdots 4\}$, $\{1 \cdot\cdot 5\}$,$\{1 \cdot\cdot 6\}$, $\{1 \cdots 7\}$

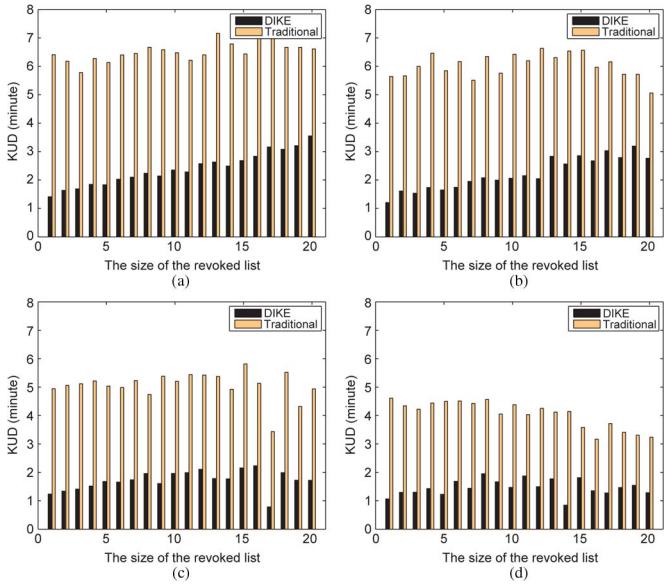


Fig. 7. Average KUD varies with the different sizes of the revoked list $|\mathcal{RL}|$ from 1 to 20. (a) RSU Group 1. (b) RSU Group 2. (c) RSU Group 3. (d) RSU Group 4.

VANET-based key distribution method, varying with different $|\mathcal{RL}|$'s. From the figure, we can see that the average KUD of the proposed DIKE scheme is much less in comparison with the traditional one. The reason is that the unidirectional communication from RSU to vehicles enables a bundle of vehicle users, although they do not drive close to the RSUs, can still receive the key update message msg, and generate their *shares* of the new session key. Only if the number of *shares* reaches the threshold τ can they cooperatively obtain the new session key.

Fig. 7 also shows that, with the increase in $\mathcal{RL}$, the average KUD in the proposed DIKE scheme will also increase, particularly when the number of deployed RSUs is small. This is because the threshold $\tau = |\mathcal{RL}| + 1$ dynamically increases with the size of the revoked list; then, when the threshold τ becomes large, it requires a vehicle user to take more time to meet more other vehicles for cooperate key update. Although the large threshold τ could increase the average KUD, we can ensure that, when the threshold value τ continues growing, the proposed DIKE scheme can still achieve lower average KUD than the traditional key distribution method since the integrated dynamic threshold technique in the proposed DIKE scheme will have no negative effect on the traditional key distribution method in VANETs. Comparing the average KUD in RSU groups 1, 2, 3, and 4, we can also observe that the

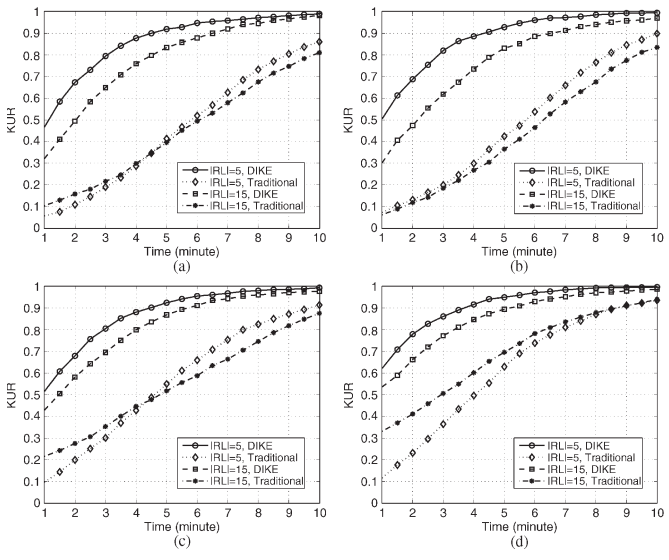


Fig. 8. Average KUR varies with time from 1 to 10 min, with $|\mathcal{RL}| = 5, 15$. (a) RSU Group 1. (b) RSU Group 2. (c) RSU Group 3. (d) RSU Group 4.

more RSUs are deployed, the lower the average KUD in both the proposed DIKE scheme and the traditional key distribution method. The reason is that, when the RSUs are deployed at more locations, many vehicle users can directly obtain new service session key with V-2-I communication. In addition, by further observing the average KUD in group 4, the average KUDs in the proposed DIKE scheme are almost same, i.e., the threshold $\tau = |\mathcal{RL}| + 1$ has little effect on the average KUD when more RSUs are deployed.

2) Average KUR: Fig. 8 shows that the average KUR varies with time from 1 to 10 min, with $|\mathcal{RL}| = 5, 15$. From the comparisons between the proposed DIKE scheme and the traditional key distribution method, we can see that the average KUR in the former always remains higher than that in the latter at any time. In addition, by comparing the average KURs in RSU groups 1, 2, 3, and 4, we can observe that the amount of deployed RSUs also has a significant impact on the average KUR. The more the RSUs deployed, the more the vehicle users that can directly obtain the service session key from RSUs; thus, the average KURs will increase quickly, particularly in the early stage. Furthermore, comparing the average KURs with $|\mathcal{RL}| = 5, 15$ in the proposed DIKE scheme, we can see that the small $|\mathcal{RL}|$ has higher KUR than the large $|\mathcal{RL}|$. The reason is that, when $|\mathcal{RL}|$ is small, the dynamical threshold τ is low; thus, a vehicle only takes less time to meet vehicles for cooperative key update. In summary, the proposed DIKE scheme, due to the involved dynamic threshold technique, can achieve better performance than the traditional key distribution method for VANETs in terms of the average KUD and KUR.

VII. Related Works

Research in dynamic key management for achieving forward secrecy and backward secrecy has been quite active in recent years. In general, these existing dynamic key management schemes can be classified into two categories, i.e., *contributory* [16]–[18] and *centralized* [19]–[21] schemes.

Contributory schemes are characterized by the lack of a group controller responsible for generating and distributing session keys for all group members [16]–[18]. Instead, all group members cooperate with each other to agree with a shared session key, and thus, it can avoid the key control attack caused by the compromise of the group controller. However, due to the distributed key generation, contributory schemes require expensive cryptographic operations, which could incur long KUD. Therefore, many contributory schemes have been proposed, aiming for improving the efficiency of key establishment. For example, Mao *et al.* [16] proposed a join-exit-tree key management framework for contributory key management, where the following holds: 1) A special key tree topology with join and exit subtrees is introduced to handle key updates for dynamic membership. 2) Optimization techniques are also employed to determine the capacities of join and exit subtrees for achieving the best time efficiency. 3) Algorithms used for dynamic updating of the join and exit trees are also designed. The analysis result shows that the asymptotic time cost for each member joining/departure event can be reduced to $O(\log(\log n))$, where n is the group size. Nevertheless, these contributory schemes are still not suitable for LBSs in VANETs since they do not consider the group member's privacy preservation. More importantly, the sparse characteristic of VANETs could make the KUD in these contributory schemes intolerable.

Centralized schemes are characterized by a TA generating and distributing a unique session key for all group members [19]–[21]. Therefore, compared with those contributory schemes, the centralized schemes are much more efficient in terms of low KUD. However, when applying these centralized key management schemes in a sparse network, how to improve the key update efficiency is still a challenging issue. In [19], Zhou and Hass first addressed the problem of how to establish a secure key management service in an ad hoc network and implemented a feasible prototype of key management scheme with threshold cryptography. In [20], Luo *et al.* proposed a distributed certification authority (dCA) with probabilistic freshness solution called DICTATE to manage the dCA, which ensures that the update is efficient at each dCA in ad hoc networks. Recently, to reduce the KUD in the MANETs, Zhang *et al.* [21] proposed a novel ID-based dynamic key management (IKM) scheme, which is closely related to our proposed DIKE scheme. IKM adopts the threshold technique under ID-based settings and cannot only ensure high-level tolerance to node compromise but can enable efficient networkwide key update via a single broadcasted message as well. However, it is still not a good candidate for dynamic key management in LBSs in VANETs since the privacy-preserving issue is not considered and the adopted threshold is static, and the secret-sharing parameters should be carefully chosen for achieving desirable levels of security and robustness.

Different from the aforementioned works, our proposed DIKE scheme is a special-purpose centralized dynamic key management for the LBSs in sparse VANETs and focuses on fast and secure LBS session key update while guaranteeing privacy preservation for vehicle users. Concretely, the proposed DIKE scheme applies the PPA technique to achieve the vehicle user's privacy-preserving authentication and uses the forward

secrecy technique [25] and dynamic threshold technique [26] to improve the key update efficiency.

VIII. Conclusion

In this paper, we have proposed a dynamic privacy-preserving key management scheme (DIKE) for securing LBSs in VANETs. Based on the unidirectional communication characteristic from RSUs to vehicles, the proposed DIKE scheme provides a new cooperative key update alternative by integrating a novel dynamic threshold technique [26] with the traditional V-2-V communications. Compared with the traditional key update policies in sparse VANETs, the proposed DIKE scheme has been identified to not only significantly reduce the KUD due to the user departure event but also to achieve the vehicle user's privacy preserving, the session key's forward secrecy and backward secrecy, and resist possible collusion from the departed vehicle users as well. In addition, through extensive performance evaluations, we have further demonstrated that the proposed DIKE scheme can achieve much better efficiency in terms of the average KUD and average KUR during each key update procedure.

References

[1] F. Martinez, C.-K. Toh, J.-C. Cano, C. Calafate, and P. Manzoni, "Emergency services in future intelligent transportation systems based on vehicular communication networks," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 2, pp. 6–20, 2010.

[2] H. Zhu, R. Lu, X. Lin, and X. Shen, "Security in service-oriented vehicular networks—Service-oriented broadband wireless network architecture," *IEEE Wireless Commun.*, vol. 16, no. 4, pp. 16–22, Aug. 2009.

[3] N. M. Drawil and O. Basir, "Intervehicle-communication-assisted localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 678–691, Sep. 2010.

[4] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[5] *Car 2 Car Communication Consortium*. [Online]. Available: http://www.car-to-car.org

[6] H. Xie, L. Kulik, and E. Tanin, "Privacy-aware traffic monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 61–70, Mar. 2010.

[7] R. Lu, X. Lin, H. Zhu, and X. Shen, "Spark: A new vanet-based smart parking scheme for large parking lots," in *Proc. IEEE 28th Conf. Comput. Commun.*, 2009, pp. 1413–1421.

[8] G. Yan, W. Yang, D. Rawat, and S. Olariu, "Smartparking: A secure and intelligent parking system," *IEEE Intell. Transp. Syst. Mag.*, vol. 3, no. 1, pp. 18–30, 2011.

[9] S. Smaldone, L. Han, P. Shankar, and L. Iftode, "Roadspeak: enabling voice chat on roadways using vehicular social networks," in *SocialNets: Proc. 1st Workshop Social Netw. Syst.*, 2008, pp. 43–48.

[10] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Security*, vol. 15, no. 1, pp. 39–68, Jan. 2007.

[11] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "AMOEBA: Robust location privacy scheme for vanet," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1569–1589, Oct. 2007.

[12] J. Freudiger, M. H. Manshaei, J.-Y. Le Boudec, and J.-P. Hubaux, "On the age of pseudonyms in mobile ad hoc networks," in *Proc. INFOCOM*, 2010, pp. 1577–1585.

[13] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.

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

[15] O. A. Omitaomu, A. R. Ganguly, B. W. Patton, and V. A. Protopopescu, "Anomaly detection in radiation sensor data with application to transportation security," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 324–334, Jun. 2009.

[16] Y. Mao, Y. Sun, M. Wu, and K. J. R. Liu, "JET: Dynamic join-exit-tree amortization and scheduling for contributory key management," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 1128–1140, Oct. 2006.

[17] M. Steiner, G. Tsudik, and M. Waidner, "CLIQUES: A new approach to group key agreement," in *Proc. ICDCS*, 1998, pp. 380–387.

[18] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 8, pp. 769–780, Aug. 2000.

[19] L. Zhou and Z. J. Hass, "Securing ad hoc networks," *IEEE Netw.*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.

[20] J. Luo, J.-P. Hubaux, and P. T. Eugster, "DICTATE: Distributed certification authority with probabilistic freshness for ad hoc networks," *IEEE Trans. Dependable Secure Comput.*, vol. 2, no. 4, pp. 311–323, Oct.–Dec. 2005.

[21] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing mobile ad hoc networks with certificateless public keys," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 4, pp. 386–399, Oct.–Dec. 2006.

[22] *Draft Amendment for Wireless Access in Vehicular Environments (Wave)*, IEEE Std. 802.11p/d5.0, Nov. 2008.

[23] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in *Proc. ACM Conf. Comput. Commun. Security*, 2004, pp. 168–177.

[24] J. R. Douceur, "The sybil attack," in *Proc. IPTPS*, 2002, pp. 251–260.

[25] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "A novel anonymous mutual authentication protocol with provable link-layer location privacy," *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 1454–1466, Mar. 2009.

[26] C. Delerablée and D. Pointcheval, "Dynamic threshold public-key encryption," in *Proc. CRYPTO*, 2008, pp. 317–334.

[27] Dedicated short range communications (DSRC). [Online]. Available: http://www.standards.its.dot.gov/Documents/advisories/dsrc_advisory.htm

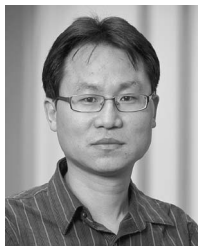
[28] H. Shan, W. Zhuang, and Z. Wang, "Distributed cooperative MAC for multi-hop wireless networks," *IEEE Commun. Mag.*, vol. 47, no. 2, pp. 126–133, Feb. 2009.

[29] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Proc. ASIACRYPT*, 2005, pp. 515–532.

[30] L. Kleinrock, *Queueing Systems Vol. 1: Theory*. Hoboken, NJ: Wiley, 1975.

Rongxing Lu (S'09–M'11) is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

He is currently a Research Assistant with the Broadband Communications Research Group, University of Waterloo. Currently, he maintains the "Bibliography on Secure Vehicular Communications" http://bbcr.uwaterloo.ca/~rxlu/sevecombib.htm with his colleagues, which has collected all vehicular ad hoc network (VANET) security-related work and become a very important Web resource for researchers around the world who are interested in security and privacy issues in VANETs. His research interests include wireless network security, applied cryptography, and trusted computing.



Xiaodong Lin (S'07–M'09) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1998 and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2008.

He is currently an Assistant Professor of information security with the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON. His research interests include wireless network security, applied cryptography, computer forensics, and software security.

Dr. Lin was the recipient of a Natural Sciences and Engineering Research Council of Canada Graduate Scholarships and the Outstanding Achievement in Graduate Studies Award during his Ph.D. studies.

Xiaohui Liang (S'10) is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

He is currently a research Assistant with the Broadband Communications Research Group, University of Waterloo. His research interests include wireless network security, applied cryptography, and e-healthcare systems.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the B.Sc. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees in electrical engineering from Rutgers University, Camden, NJ, in 1987 and 1990, respectively.

He is currently a Professor and University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is an Editor-in-Chief for *Peer-to-Peer Networks and Applications*. His research interests include mobility and resource management, ultrawideband wireless networks, wireless network security, and vehicular ad hoc and sensor networks.

Prof. Shen is a registered Professional Engineer in Ontario, Canada. He is an Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He is a Fellow of the Engineering Institute of Canada and a Distinguished Lecturer of the IEEE Communications Society.