# On the Security of a Privacy-Preserving Key Management Scheme for Location Based Services in VANETs

Bao Liu[1], Lei Zhang[1⋆], Josep Domingo-Ferrer[2]

[1]Shanghai Key Laboratory of Trustworthy Computing
Software Engineering Institute, East China Normal University, Shanghai, China
[2]UNESCO Chair in Data Privacy
Department of Computer Engineering and Mathematics
Universitat Rovira i Virgili, Tarragona, Catalonia
baoliu@ecnu.cn, leizhang@sei.ecnu.edu.cn, josep.domingo@urv.cat

**Abstract.** Location based services (LBSs) are promising value-added services in vehicular *ad hoc* networks (VANETs), which can yield substantial economic profits. To extensively deploy LBSs in VANETs, it is essential to establish an efficient privacy-preserving key management scheme. In this paper, we point out a privacy weakness in a recent key management scheme based on group signatures for LBSs in VANETs; then we propose a secure and privacy-enhanced version. In our scheme, roadside units (RSUs) act as group managers. Vehicles are distributed into groups maintained by these RSUs. If a vehicle's member key is compromised, one just needs to update the group public key corresponding to its group manager. With this method, the member revocation and privacy leakage problems in schemes based on group signatures are solved effectively. As a result, a vehicle may enjoy LBSs efficiently without surrendering its privacy.

**Key words**: Security, Vehicle privacy, Vehicular *ad hoc* networks, Location based services, Key management.

## 1 Introduction

As the society is paying more and more attention to the road safety and efficiency, vehicular *ad hoc* networks (VANETs) are becoming a hot research spot. Based on the vehicle-to-vehicle (V2V), vehicle-to-roadside (V2R) and roadside-to-vehicle (R2V) communications in VANETs, vehicles can communicate with each other and the nearby RSUs to share information about the current traffic conditions (e.g., traffic jams, accidents, icy roads, flooded roads, closed roads, etc). With these mechanisms, VANETs are expected to improve road safety and traffic efficiency. In the near future, VANETs are expected to serve as a general platform for the development of any vehicle centered application [9].

⋆ Corresponding author.

In addition to the safety applications, value-added applications are envisioned to offer various entertaining services to drivers and passengers, e.g., internet access, navigation, social media, payTV, etc. Obviously, value-added applications promise substantial business opportunities. Location based service (LBS) is one type of promising and important value-added applications. To deploy LBSs in VANETs, one must deal with security issues (e.g., identity authentication, data integrity) and privacy issues (e.g., protection of the user's identity and/or location). Further, since LBSs are only intended for authenticated vehicles, we also have to consider confidentiality. A substantial body of studies has been devoted to security, privacy and confidentiality issues in VANETs (e.g., [4,8,9,10,17,18,19]). However, only a few of them (e.g., [8,9]) concern the key management for LBSs, which is crucial for efficient confidential communications in LBS sessions.

In [9], a dynamic privacy-preserving key management scheme was designed for location based services in VANETs. In this scheme, a privacy-preserving authentication (PPA) scheme, which is derived from the verifier-local group signature scheme in [2], is proposed for a vehicle to securely obtain a session key from a service provider without violating its privacy. We note that member revocation is usually a crucial problem in group signature based schemes. In [9], all the vehicles (members) are in a single group (which may contain numerous vehicles) maintained by a trusted authority (TA). It is reasonable to assume that a vehicle's member key may be leaked. Therefore, we need to consider the member revocation problem.

### 1.1 Contribution and Plan of this Paper

In this paper, we first point out a privacy weakness in the key management scheme for location based services in [9]. Although the PPA scheme can be used to protect the privacy of a vehicle, if a vehicle's member key is leaked, its privacy may be under threat. This is because when the member key of a vehicle is revoked, the revocation token of the vehicle should be added to the revocation list maintained by the TA. Based on the revocation token, a service provider can link all the requests of the revoked vehicle, which results in undesirable vehicle profiling. Therefore, it is reasonable to consider vehicle privacy even after revocation.

Secondly, we propose a secure and privacy enhanced version. In our scheme, TA never acts as the group manager. Instead, RSUs work as group managers to manage the groups. Before joining an LBS session, a vehicle has to request a member key from an RSU at first. The TA is only used to authenticate a vehicle and to detect whether a vehicle has already obtained a member key from an RSU. In this way, the vehicles are enrolled into various groups maintained by the RSUs. To solve the member revocation problem, instead of using revocation lists, we assume the RSUs can update their public keys. Once a vehicle is revoked, its corresponding group manager (RSU) has to update the group public key. We note that the number of vehicles enrolled by a single RSU may not be large and the vehicles in this group who are in an LBS session are limited. The public key

update mechanism influences the efficiency of the whole system only slightly. Further, batch verification mechanism is used to improve the efficiency of the scheme.

The rest of this paper is organized as follows. Section 2 introduces the system architecture and the security requirements. In Section 3, we show some technical preliminaries utilized in our scheme. Section 4 reviews the key management scheme in [9] and points out a weakness in it. In Section 5, we propose a privacy enhanced version. Section 6 evaluates the proposed scheme. Section 7 concludes this paper.

## 2 Background

In this section, we illustrate the system architecture and the security requirements for LBSs in VANETs.

### 2.1 System Architecture

Figure 1 shows the system architecture in [9] and in this paper. It consists of a trusted authority (TA), roadside units (RSUs), service providers (SPs) and vehicles.

◇ **TA** : It is a trusted third party who generates public parameters for the whole system. All entities in the system must register with TA to get their own private and public keys. TA is also employed to keep and manage the real identity of each vehicle.

◇ **SP** : Service providers that provide LBSs, such as establishing an exclusive network for the vehicles from the same corporation or building a social network for the vehicles with the same destination based on their location.

◇ **RSU** : The road-side units are deployed along the roadside, and they are always equipped with some processing units and sensors. In our scheme, RSUs are used to maintain groups and transfer messages between vehicles and TA/SPs.

◇ **Vehicle** : Vehicles are the customers of LBSs and they are all equipped with on-board units (OBUs). With OBUs, vehicles are able to communicate with the other entities.

### 2.2 Security Requirements

In this section, we illustrate the security requirements for LBSs in VANETs.

– **Message Confidentiality**. An LBS request may contain sensitive information. Therefore, message confidentiality is required. This means that only the vehicle and the RSU/SP should see the information exchanged between the vehicle and the RSU/SP. In addition, a vehicle should not be able to obtain the services provided by the SP before it joins the session or after it departs from the session. This also implies the *forward secrecy* and the *backward secrecy* properties defined in [9].

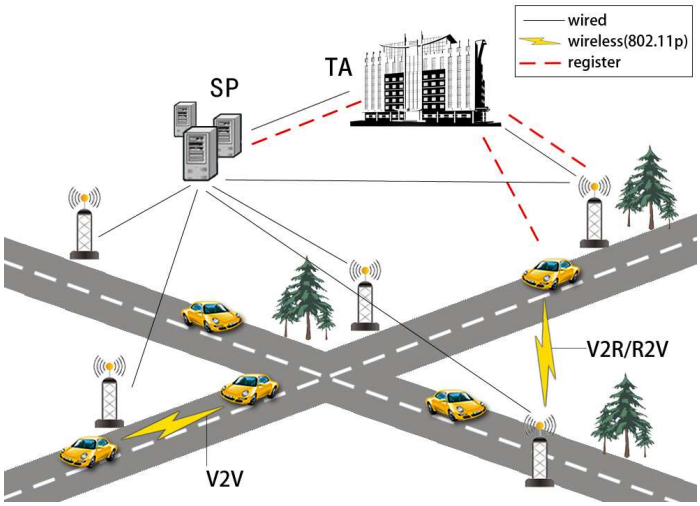


**Fig. 1.** System Architecture

- **Vehicle Authentication**. The SP wants to ensure that the LBS is only provided to its subscribers. In addition, the SP should be able to detect whether the vehicle is double-registering to the same session, in order to avoid some attacks (e.g., the Sybil attack [6]).
- **Vehicle Privacy**. Except the message generator and TA, it must be computationally hard for any entity to decide whether two different messages are generated by the same vehicle. For LBSs to gain wide social acceptance, it is essential to protect the privacy of vehicles.

## 3 Technique Preliminaries

In this section, we explicate some preliminaries served as bases in our scheme. Mainly, they are bilinear maps and group signatures.

### 3.1 Bilinear Maps

Many efficient cryptosystems are based on bilinear maps. Our scheme is also built on them. Therefore, we briefly review them here.

Let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be three multiplicative cycle groups of the same prime order $q$. Let $g_1$ and $g_2$ be the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Let $\Psi$ denote a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, specifically, $\Psi(g_2) = g_1$. A map $e$: $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ is called bilinear map if it satisfies $e(g_1, g_2) \neq 1$ and $e(u^a, v^b) = e(u, v)^{ab} \in \mathbb{G}_T$, where $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in {\mathbb{Z}_q}^*$. In the rest of this paper, we call the tuple $\{q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e\}$ the basic bilinear map parameters.

### 3.2 Group Signatures

Group signatures allow any member of the group to produce a signature on behalf of the group. No one can know the real identity of the signer except the group manager, who issues member keys for the members. Furthermore, it is computational hard for anyone but the manager to distinguish whether two signatures are issued by the same signer.

Member revocation is a crucial problem in group signature based schemes. That is, if a member leaves the group or its group member key is leaked, the problem is how to exclude this member from the group. Verifier-local revocation (VLR) group signatures are designed to alleviate this member revocation problem. In a VLR group signature scheme, each member has a revocation token. If a member is revoked, the group manager just needs to publish its revocation token. The other members are able to distinguish the revoked member based on the revocation token.

## 4 Privacy Weakness of a Key Management Scheme

In this section, we first review the key management scheme in [9] at high level and then point out a privacy weakness of the scheme.

### 4.1 High Level Description

There are four stages in the key management scheme in [9]: system initialization, LBS settings, vehicle joining and vehicle departure. We outline these stages as follows.

In the first stage, TA generates the system master key and the public parameters of the system. It also extracts a private key for each SP. Furthermore, TA also acts as a group manager of a group signature scheme who issues member keys for the vehicles in the system.

During the second stage, each SP initializes the parameters for the LBS sessions. The SP also generates a series of keys(e.g., the session master key, the session key, the combine keys and the encryption keys), beacon messages and a dummy user set. The session key, the combine keys and the encryption keys are derived from the session master key. The session key will be distributed to the authenticated vehicle, and the combine keys and the encryption keys will be used to update the session key in the last stage. The beacon message is broadcasted periodically via RSUs to inform the vehicles of the service. The dummy user set will be used to help a vehicle to update the session key when it cannot connect to the SP.

In the third stage, if a vehicle wants to join an LBS session, it first verifies the beacon message broadcasted by the corresponding SP. If the beacon message is valid, the vehicle will request to join the session with a privacy-preserving authentication (PPA) scheme, which is derived from the verifier-local group signature scheme in [2]. By the PPA scheme, the SP can authenticate vehicles

without leaking their private information and is able to check the vehicle's double registration [9]. If a vehicle is authenticated as valid, the SP generates a unique pseudonym and the private key corresponding to the pseudonym for the vehicle, which will be used for the session key update. Finally, the SP sends pseudonym, private key, session key and combine key to the vehicle through a secure channel. With the session key, the session members are able to access the LBS.

The last stage is vehicle departure. A vehicle may leave the session. Therefore, it is reasonable for the SP to update the session key to avoid the vehicle accessing the service after its departure. To do this, when a vehicle leaves the LBS session, a key update message is generated by the SP and broadcasted via RSUs to the session members. In the case that a vehicle can connect to the SP through one hop or multi-hop communication via an RSU, then the vehicle can get the new session key from the SP directly; otherwise, the vehicle derives the new session key by a dynamic threshold update algorithm [5], which takes as input pseudonym, private key, current session key, encryption key, combine key, and dummy user set.

### 4.2 The Weakness

In the above scheme, the PPA scheme (which is essentially a verifier-local group signature scheme) is adopted to secure the system. However, as mentioned in Section 3.2, the group manager needs to publish the revocation token of a revoked vehicle. This leads to a new problem. If a revocation token is published, the signatures related to this revocation token can be linked. Therefore, if a vehicle in the system is revoked, its privacy may be violated. Furthermore, the efficiency of the system declines as the scale of the revocation list grows.

## 5 Privacy Enhanced Key Management Scheme

In this section, we propose a privacy-enhanced scheme. To simplify the description, we first define some notations in Table 1.

### 5.1 The Scheme

As the scheme in [9], our scheme also has four stages: system initialization, LBS settings, vehicle joining and vehicle departure.

**[System initialization]**

The first stage is similar to that of [9]. In this stage, TA generates its master key and the public parameters. It also generates private keys for SPs. Unlike in [9], TA does not issue group member keys for vehicles, but leaves this task to the RSUs. For a vehicle, it only issues a secret, which is used to authenticate the vehicle. In addition, each RSU in our system has a private-public key pair and a certificate issued by TA.

**Table 1.** Description of notation

| NOTATION | Description. |
|---|---|
| TA | The trusted authority |
| $V$ | A vehicle. |
| $R$ | An RSU or its corresponding group. |
| SP | An LBS service provider. |
| $T_t$ | A time stamp. |
| $\|\|$ | Message concatenation operation. |
| $ID$ | The identity of an entity. |
| PUB | The public parameters. |
| $SK$ | The secret key. |
| $PK$ | The public key. |
| $mk$ | The group member key. |
| CRL | The certificate revocation list maintained by TA. |
| RL | The record list maintained by TA. |
| ML | The member list maintained by TA. |
| $E_K(\cdot)/D_K(\cdot)$ | A symmetric encryption/decryption scheme. |

Let the system master key be $\gamma \in {\mathbb{Z}_q}^*$, the public parameters PUB= $(q, g_1, g_2, u_1, u_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, E_K(\cdot)/D_K(\cdot), H, H_0, H_1, H_2, l)$, where $\{q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e\}$ is a tuple of basic bilinear map parameters, $H : \{0,1\}^* \longrightarrow {\mathbb{Z}_q}^*$, $H_0 : \{0,1\}^* \longrightarrow {\mathbb{G}_2}^2$, $H_1 : \mathbb{G}_T \longrightarrow \{0,1\}^l$, $H_2 : \{0,1\}^* \longrightarrow \mathbb{G}_2$ are cryptographic hash functions, $u_1 = {g_1}^\gamma$, $u_2 = {g_2}^\gamma$, $E_K(\cdot)/D_K(\cdot)$ represents a symmetric key encryption/decryption scheme and $l$ is the length of the key used in the encryption/decryption scheme. For an RSU $R_j$, TA selects $\zeta_j \in {\mathbb{Z}_q}^*$ as its private key and it computes $PK_{R_j} = {g_2}^{\zeta_j}$ as its public key. Finally, TA issues a certificate $Cer_{R_j}$ of the form $\{ID_{R_j}, PK_{R_j}, T, Sig_{R_j}\}$ [11] for $R_j$, where $ID_{R_j}$ is the unique identity of $R_j$, $T$ is the certificate lifetime, $Sig_{R_j}$ is TA's signature on $(ID_{R_j}, PK_{R_j}, T)$. Each RSU broadcasts its certificate periodically in its communication range. For an SP, TA computes $SK_{SP} = g_1^{\frac{1}{\gamma + H(ID_{SP})}}$ as SP's private key, where $ID_{SP}$ is the identity of the SP. For a vehicle $V_k$, TA randomly selects $x_k \in \{0,1\}^\ell$ as $V_k$'s secret which will be used to authenticate a vehicle, where $\ell$ is a security parameter.

TA also manages three lists: certificate revocation list (CRL), member list (ML) and record list (RL). The CRL contains the information of the revoked certificates of RSUs. ML has the form $(ID_{V_k}, x_k)$, where $ID_{V_k}$ is the real identity of $V_k$. RL will be defined in the third stage.

**[LBS settings]**

This stage is the same as that of the scheme in [9]. The SP generates the initial session key, a dummy user set and a series of session related keys, which contains the session master key, the combine keys, the encryption keys. In addition, SP also constructs and broadcasts the LBS session beacon message in this stage.

**[Vehicle joining]**

In the stage of vehicle joining, if a vehicle $V_k$ wants to join an LBS session, it must join a group maintained by an RSU at first. When a vehicle $V_k$ receives the beacon message of its interested LBS, it first verifies the LBS beacon message. The verification procedure is the same as that of the scheme in [9]. If $V_k$ is not a member of any group maintained by an RSU or its member key has expired, $V_k$ registers to its nearby RSU $R_j$ using the Group-Join protocol below. Figure 2 illustrates the basic idea.

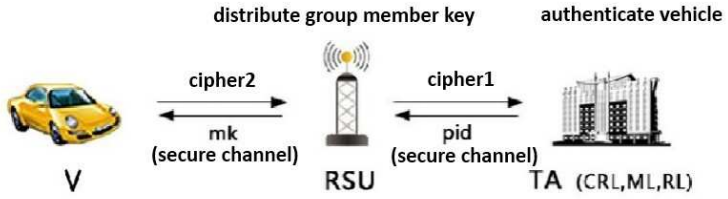


**Fig. 2.** Group-Join

### Group-Join Protocol

1. $V_k$ verifies the certificate $Cer_{R_j}$ of $R_j$. If it is valid, $V_k$ extracts the public key $PK_{R_j}$ and the identity $ID_{R_j}$ of $R_j$ from $Cer_{R_j}$, sets $M_1 = (T_t \parallel x_k)$, randomly selects $s \in \mathbb{Z}_q{}^*$, computes $C_1{=}g_1{}^s$, $K_1{=}H_1(e(u_1{}^s, u_2))$, $C_2{=}E_{K_1}(M_1)$, sets $cipher1{=}(C_1 \parallel C_2)$, chooses $f \in \mathbb{Z}_q{}^*$ at random, sets $M_2{=}(T_t \parallel ID_{R_j} \parallel cipher1 \parallel'' join'')$, computes $C_3{=}g_1{}^f$, $K_2{=}H_1(e(u_1{}^f, PK_{R_j}))$, $C_4{=}E_{K_2}(M_2)$, sets $cipher2{=}(C_3 \parallel C_4)$, sends $cipher2$ to $R_j$, where $T_t$ is a timestamp.
2. When $R_j$ receives $cipher2$, it computes $K_2 = H_1(e(C_3, u_2{}^{\zeta_j}))$, computes $M_2 = D_{K_2}(C_4) = (T_t \parallel ID_{R_j} \parallel cipher1 \parallel'' join'')$. If $T_t$ and $ID_{R_j}$ are valid, $R_j$ forwards $cipher1$ to TA.
3. When TA receives $cipher1$ from $R_j$, it computes $K_1 = H_1(e(C_1, u_2{}^{\gamma}))$, $M_1 = D_{K_1}(C_2) = (T_t \parallel x_k)$. If $T_t$ is valid, it does the following:
   - If $x_k$ does not exist in a tuple $(ID_{V_k}, x_k)$ on ML, TA replies with 0, which means that $V_k$ is not a registered vehicle.
   - Else if there exists a tuple $(ID_{V_k}, ID_{R_*}, pid_*)$ on RL, it means $V_k$ is already a member of the group maintained by $R_*$. TA replies with 1, which means that $V_k$'s member key is still valid.
   - Else, TA generates a pseudonym $pid_i$ for $V_k$, replies with $pid_i$ to $R_j$.
4. If $R_j$ receives 1 or 0 from TA, it aborts; otherwise, it randomly selects $\eta_i \in \mathbb{Z}_q{}^*$, computes $\theta_i{=}g_1{}^{\frac{1}{\zeta_j+\eta_i}}$, sets $mk_i = (\eta_i, \theta_i)$ as the group member key, adds the tuple $(mk_i, pid_i)$ to its group member list, computes $cipher3 = E_{K_2}(mk_i)$, sends $cipher3$ to $V_k$.

5. When $V_k$ receives $cipher3$ from $R_j$, it computes $D_{K_2}(cipher3)$ to extract the member key.
6. Finally, if the protocol is successfully terminated, $(ID_{V_k}, ID_{R_j}, pid_i)$ is added to RL.

In the above Group-Join protocol, only when $x_k$ is valid and the identity of the vehicle corresponding to $x_k$ is not recorded in RL, TA generates a pseudonym for $V_k$. By this mechanism, a vehicle cannot register with two groups maintained by two different RSUs. In other words, each vehicle cannot obtain more than one group member key. Therefore, similar to [9], the double-registration can be detected using the PPA scheme.

After getting $mk_i$ from $R_j$, $V_k$ can join any LBS session via a nearby RSU $R_i$ using the PPA scheme as [9]. The communication model is shown in Figure 3, where "request" represents the message sent by a vehicle to the SP for joining a session and "reply" represents all the keys that the SP distributed to the vehicle. The basic procedure is illustrated in following Session-Join Protocol. We note that the only difference between our idea and that in [9] is that the group manager of $V_k$ is no longer the TA but $R_j$. Therefore, $V_k$ also needs to send the certificate of its group manager $R_j$ to $R_i$. $R_i$ has to check the validity of the certificate when it performs the PPA scheme.

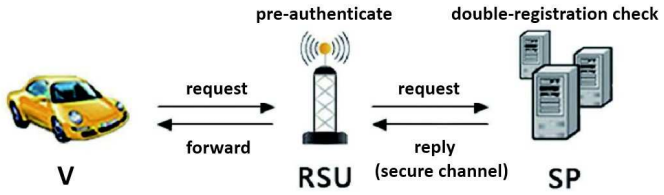


**Fig. 3.** Session-Join

### Session-Join Protocol

1. $V_k$ generates a joining session request as follows:
   (a) Computes $(\hat{u}, \hat{v}) = H_0(PUB, sid)$, $u = \psi(\hat{u})$, $v = \psi(\hat{v})$, randomly selects $\alpha \in {Z_q}^*$, sets $\delta = \eta_i \cdot \alpha$, $T_1 = u^\alpha$, $T_2 = \theta_i \cdot v^\alpha$.
   (b) Generates $R_1 = u^{r_\alpha}$, $R_2 = e(T_2, g_2)^{r_x} \cdot e(v, PK_{R_j})^{-r_\alpha} \cdot e(v, g_2)^{-r_\delta}$ and $R_3 = {T_1}^{r_x} \cdot u^{-r_\delta}$, computes $h = H(PUB, sid, ID_{R_j}, T_t \parallel {g_e}^x \parallel {g_e}^y, T_1, T_2, R_1, R_2, R_3)$, where $r_\alpha, r_x, r_\delta, y$ are randomly selected from ${Z_q}^*$, $g_e = e(g_1, g_2)$, ${g_e}^x$ is obtained from the beacon message broadcasted by the SP via RSUs.

(c) Computes $s_\alpha = r_\alpha + h \cdot \alpha$, $s_x = r_x + h \cdot \eta_i$, $s_\delta = r_\delta + h \cdot \delta$, sets the request as $Req = (ID_{R_j}, T_t \parallel {g_e}^x \parallel {g_e}^y, T_1, T_2, h, s_\alpha, s_x, s_\delta)$, sends $Req$ to its nearby RSU $R_i$.

2. When $R_i$ receives $Req$ from $V_k$, it pre-authenticates the request as follows:
   (a) Extracts $ID_{R_j}$ from the request, checks the validity of its corresponding certificate $Cer_{R_j}$ and $T_t$ in the request,
   (b) If both of them are valid, computes $R'_1 = \frac{u^{s_\alpha}}{{T_1}^h}$, $R'_2 = e(T_2, g_2)^{s_x} \cdot e(v, PK_{R_j})^{-s_\alpha} \cdot e(v, g_2)^{-s_\delta} \cdot (e(T_2, PK_{R_j})/e(g_1, g_2))^h$ and $R'_3 = {T_1}^{s_x} \cdot u^{-s_\delta}$, checks whether the following equation holds:

$$h = H(PUB, sid, ID_{R_j}, T_t || {g_e}^x || {g_e}^y, T_1, T_2, R'_1, R'_2, R'_3).$$

   (c) If the equation holds, forwards the request to the SP.

3. While receiving the request from $R_i$, the SP does the double-registration check at first. If the request is legal, it then builds a secure communication channel with $V_k$. The detailed steps comes as follows:
   (a) SP extracts $(T_1, T_2)$ from the request, checks whether the equation

$$e(T_2, \hat{u})e(T_1, \hat{v}^{-1}) = e(T_2^i, \hat{u})e(T_1^i, \hat{v}^{-1})$$

   holds for $(T_1^i, T_2^i) \in \mathrm{T} = \{(T_1^1, T_2^1), \cdots, (T_1^m, T_2^m)\}$, where $m$ represents the number of registered vehicles.
   (b) If none of the above equations holds, SP adds $(T_1, T_2)$ to T, computes the encryption key $(g_e^y)^x$, generates pseudonym, private key, session key and combine key, encrypts all the keys with $(g_e^y)^x$, and sends the encrypted message to $V_k$.

In the above Session-Join Protocol, an SP may receive amounts of requests at the same time. If it performs the double-registration check individually, then it will be quite time consuming due to the expensive bilinear map operation. We therefore put forward an efficient batch verification algorithm to improve the performance. Assume there are $N$ requests. We phrase the $n$-th request as $Req_n = (ID_{R_j}, T_{n,t} || {g_e}^x || {g_e}^{y_n}, T_{n,1}, T_{n,2}, h_n, s_{n,\alpha}, s_{n,x}, s_{n,\delta})$, where $n \in (1, N)$. The SP does the batch verification as follows:

1. For $1 \leq n \leq N$, extracts the $(T_{n,1}, T_{n,2})$ part from each request $Req_n$, selects random width-$\omega$ non-adjacent forms ($\omega$-NAFS [3]) $\delta_1, ..., \delta_N$.
2. For each $(T_1^i, T_2^i)$ in T, $1 \leq i \leq m$, checks whether the following equation holds:

$$e(\prod_{n=1}^{N} T_{n,2}^{\delta_n}, \hat{u})e(\prod_{n=1}^{N} T_{n,1}^{\delta_n}, \hat{v}^{-1}) = e(\prod_{n=1}^{N} (T_2^i)^{\delta_n}, \hat{u})e(\prod_{n=1}^{N} (T_1^i)^{\delta_n}, \hat{v}^{-1}).$$

3. If none of the above equations holds, replies the keys as that of the Session-Join Protocol. Otherwise, one may employ the recursive divide-and-conquer approach in [7] to exclude the illegal request(s).

**[Vehicle departure]**

In the last stage, similar to that of the scheme in [9], if any session member $V_k$ departs from the session, the SP needs to update the session key. This procedure is the same as that in [9]. In addition, the group member key of a vehicle $V_k$ may be leaked. In [9], if the member key of a vehicle is compromised, TA has to publish the revocation token of the vehicle. As discussed in Section 3.2, the privacy of the vehicle is violated and the efficiency of the system declines. In our system, we propose a new method to solve the member revocation problem. When a vehicle $V_k$'s member key is compromised, $V_k$ has to send a revocation request to TA. Suppose $V_k$ is a member of the group maintained by $R_j$. When TA receives this request, TA adds $Cer_{R_j}$ to CRL and it deletes all the records of the form $(*, ID_{R_j}, *)$ in RL. $R_j$ will generate a new private-public pair and receive a new certificate issued by TA. Note that, since we distribute the vehicles to many groups maintained by RSUs, the number of group members in the group maintained by $R_j$ will not be large. We also note that, by our setting, the SP has to periodically check the CRL. Once the SP finds that the certificate of an RSU corresponding to its session member(s) is revoked, it excludes all this/these session member(s) and updates the session key instantly.

## 6 Evaluation

In this section, we evaluate the proposed key management scheme. Firstly, we show that our scheme meets all the security requirements defined in Section 2.2. Then, we evaluate the performance of the proposed scheme.

### 6.1 Security Analysis

The key management scheme in [9] is shown to satisfy message confidentiality, vehicle authentication and vehicle privacy as defined in Section 2.2. Our scheme is based on the scheme in [9]. The main difference between our key management scheme and that in [9] is that our scheme also considers the privacy of a vehicle when it is revoked. This is realized by distributing the vehicles into the groups maintained by RSUs and updating the public key of an RSU when its member is revoked. We only need to consider the security of our Group-Join Protocol. In the Group-Join protocol, the symmetric key encryption/decryption scheme is employed to guarantee the secrecy of the massages. Hence, confidentiality is satisfied. Further, in the protocol, a vehicle may obtain a member key if it is authenticated by TA. Except TA, no entity can learn the real identity of a vehicle. Hence, vehicle authentication and vehicle privacy are achieved.

### 6.2 Performance Evaluation

The main differences between our scheme and that in [9] are the group joining and session joining protocols, in which the latter mainly dominates the efficiency of the whole scheme. We thus compare the performance of session joining protocol in our scheme with that in [9].

The most time consuming operation in the scheme is the bilinear map operation. Let $\tau_m$ denote the time to compute a bilinear map. We select a non-supersingular curve with an embedded degree $k = 6$ and run the the computation on an Intel i5-2430M 2.4-GHz machine[1], $\tau_m$ is 2.17 ms.

In our proposed scheme, a vehicle has to join an RSU maintained group first and then join the session. For a single vehicle, the total time cost of this whole joining procedure is similar to that of [9]. However, under the condition that many vehicles request to join a session at the same time, our scheme will be more efficient. Assume there are $N$ vehicles request to join a session at the same time. In the best case, i.e., all the requests are legal, our scheme only takes $2(m + 1)\tau_m$ to do the double-registration check, while the scheme in [9] takes $2N(m + 1)\tau_m$, where $m$ represents the number of registered vehicles. According to [7], even if up to 15% requests are invalid, the performance of our scheme is better than the individual check.

## 7 Conclusion

We pointed out a privacy weakness of a recent key management scheme for LBSs in VANETs and proposed a privacy enhanced one. In our scheme, each vehicle can join a group maintained by an RSU without leaking its privacy. Furthermore, each vehicle can be authenticated without leaking its privacy when it joins an LBS session. We also proposed a novel method to solve the member revocation problem in group signature based systems. By our method, even if a vehicle's group member key is compromised, its privacy still can be protected.

**Acknowledgments and disclaimer**

Thanks goes to Chuanyan Hu and Ya Gao for the proofreading. This work was supported in part by the NSF of China under grants 61202465, 61021004, 11061130539, 91118008 and 61103222; EU FP7 under projects "DwB" and "Inter-Trust"; the Spanish Government under projects TIN2011-27076-C03-01 and CONSOLIDER INGENIO 2010 "ARES" CSD2007-0004; the Government of Catalonia under grant SGR2009-1135; the Shanghai NSF under grant no. 12ZR1443500; the Shanghai Chen Guang Program (12CG24); the Fundamental Research Funds for the Central Universities of China; the Open Project of Shanghai Key Laboratory of Trustworthy Computing (no. 07dz22304201101). J. Domingo-Ferrer was supported in part as an ICREA-Acadèmia researcher by the Government of Catalonia.

## References

1. Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL). http://www.shamus.ie/

[1] The operation system is Ubuntu 12.04 and exploiting the Miracl library [1].

2. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: 11th ACM Conference on Computer Communications Security-CCS 2004, pp. 168-177(2004)
3. Cheon, J., Yi, J.: Fast batch verification of mulitiple signatures. In: Public-Key Cryptography-PKC 2007. LNCS, vol. 4450, pp. 442-457. Springer Berlin Heidelberg(2007)
4. Chim, T. W., Yiu, S. M., Hui, L. C. K., Li, V. O. K.: Security and privacy issues for inter-vehicle communications in VANETs. In: Procedings of 6th Annual IEEE Communications Society Conference on SECON Workshops, pp. 1-3(2009)
5. Delerablée, C., Pointcheval, D.: Dynamic threshold public-key encryption. In: Advances in Cryptology-CRYPTO 2008. LNCS, vol. 5157, pp. 317-334. Springer Berlin Heidelberg(2008)
6. Douceur, J. R.: The Sybil attack. In: Proceedings of Peer-to-Peer Systems-IPTPS 2002. LNCS, vol. 2429, pp. 251-260. Springer Berlin Heidelberg(2002)
7. Ferrara, A. L., et al.: Prictical short signature batch verigfication. In: CT-RSA 2009. LNCS, vol. 5473, pp. 309-324. Springer Berlin Heidelberg(2009)
8. Huang, J.-L., Yeh L.-Y., Chien, H.-Y.: ABAKA: an anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology, 60(1), 248-262(2011)
9. Lu, R., Lin, X., Liang X., Shen, X.: A dynamic privacy-preserving key management scheme for location based services in VANETs. IEEE Transactions on Intelligent Transportation Systems, 13(1), 127-139(2012)
10. Mahajan, S., Jindal, A.: Security and privacy in VANET to reduce authentication overhead for rapid roaming networks. International Journal of Computer Applications, 1(20), 21-25(2010)
11. Papadimitratos, P., Buttyan, L., Hubaux, J., Kargl, F., Kung, A., Raya, M.: Architecture for secure and private vehicular communications. In: 7th International Conference on Intelligent Transportation Systems-ITS 2007, pp. 1-6(2007)
12. Raya, M., Aziz, A., Hubaux, J.: Efficient secure aggregation in VANETs. In: Proceedings of the 3rd International Workshop on Vehicular *Ad Hoc* NetWorks 2006, pp. 67-75(2006)
13. Raya, M., Hubaux, J.: The security of vehicular *ad hoc* networks. In: 3rd ACM Workshop on Security of *Ad hoc* and sensor networks-SASN 2005, pp. 11-21(2005)
14. Wu, Q., Domingo-Ferrer, J., González-Nicolás, U.: Balanced trustworthiness, safety and privacy in vehicle-to-vehicle communications. IEEE Transactions on Vehicular Technology, 59(2), 559-573(2010)
15. Wu, Q., Qin, B., Zhang, L., Domingo-Ferrer, J., Farras, O.: Bridging Broadcast Encryption and Group Key Agreement. In: 17th Annual International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2011). LNCS, vol. 7073, pp. 143-160. Springer Berlin Heidelberg(2011)
16. Zhang, C., Lu, R., Lin, X., Ho, P.-H., Shen, X.: An efficient identity-based batch verification scheme for vehicular sensor networks. In: 27th Conference on Computer Communications, pp. 246-250(2008)
17. Zhang, L., Wu, Q., Qin, B., Domingo-Ferrer, J.: Practical Privacy for Value-Added Applications in Vehicular Ad Hoc Networks. In: 5th International Conference on Internet and Distributed Computing Systems (IDCS 2012). LNCS, vol. 7646, pp. 43-56. Springer Berlin Heidelberg(2012)
18. Zhang, L., Wu, Q., Qin, B., Domingo-Ferrer, J.: APPA: Aggregate Privacy-Preserving Authentication in Vehicular Ad Hoc Networks. In: 14th Information Security Conference (ISC 2011). LNCS, vol. 7001, pp. 293-308. Springer Berlin Heidelberg(2011)

19. Zhang, L., Wu, Q., Solanas, A., Domingo-Ferrer, J.: A scalable robust authentication protocol for secure vehicular communications. IEEE Transactions on Vehicular Technology 59(4), 1606-1617(2010)