Practical Privacy for Value-Added Applications in Vehicular *Ad Hoc* Networks

Lei Zhang[1], Qianhong Wu[2,3], Bo Qin[2,4], and 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, Dept. of Comp. Eng. and Maths
Universitat Rovira i Virgili, Tarragona, Catalonia
[3] Key Lab. of Aerospace Information Security and Trusted Computing
Ministry of Education, Wuhan University, School of Computer, China
[4] Department of Maths, School of Science, Xi'an University of Technology, China
`leizhang@sei.ecnu.edu.cn, {qianhong.wu,bo.qin,josep.domingo}@urv.cat`

Abstract. Advances in mobile networking and information processing technologies have triggered vehicular *ad hoc* networks (VANETs) for traffic safety and value-added applications. Most efforts have been made to address the security concerns while little work has been done to investigate security and privacy for value-added applications in VANETs. To fill this gap, we propose a value-added application, specifically, a security and privacy preserving location-based service (LBS) scheme for VANETs. For each LBS transaction, the scheme provides authentication, integrity and non-repudiation for both the service provider and the user. A user can obtain the service in an anonymous way and hence user privacy is well protected. However, a tracing procedure can be invoked to find malicious users, thereby efficiently preventing users from abusing the anonymity provided by the system.

Keywords: Information security, Vehicular *ad hoc* networks, Location based service, Conditional privacy.

1 Introduction

Vehicular *ad hoc* networks (VANETs) consist of computers mounted on vehicles and road infrastructures, and they are emerging as the first commercial mobile *ad hoc* networks. This kind of networks allow vehicle-to-vehicle and vehicle-to-roadside communications by means of on-board units (OBUs) installed in each vehicle as well as roadside units (RSUs) deployed alongside roads. The development of VANETs is expected to improve traffic safety and management efficiency by allowing information on current traffic conditions to be shared in quasi-real time; obvious benefits will be driver assistance, traffic management, and handling of traffic jams and emergencies. To this end, a large body of proposals (e.g. [7,10,15,16]) have been proposed to guarantee trustworthiness of vehicle-generated messages and address the security concerns in VANETs.

In addition to safety applications, VANETs may enable a broad range of value-added applications. Among them, LBSs are expected to open substantial business opportunities. However, to let LBSs be widely deployed in VANETs, specific security and privacy requirements have to be met. For instance, any LBS user should be authentic and any transaction must be non-repudiable to guarantee that providing LBSs is profitable. Also, user privacy[1] should be ensured against a malicious LBS provider to prevent maliciously tracing or profiling users. In traditional wired networks, sophisticated cryptographic technologies have been developed to protect parties in value-added applications including LBSs. However, all these protocols are only suitable for a full connectivity scenario. VANETs are very dynamic and their communications are volatile, which makes those complex protocols unsuitable.

Related Work. In recent years, a number of papers have dealt with security and privacy in VANETs [7,10,15,16]. However, few efforts have been made to address the security and privacy issues of value-added applications attracting drivers to use VANETs. To fill this gap, Sampigethaya *et al.* [12] proposed a scheme called AMOEBA. In their scheme, the group concept is introduced to provide anonymous access to LBSs and prevent a malicious service provider from profiling any target vehicle. However, as remarked by the authors, there are several limitations left unsolved in their scheme. For instance, due to the dynamic and volatile connections in VANETs, the groups in this kind of networks might be hard to maintain. Even if the group can remain, the privacy of the group leader is sacrificed to achieve privacy for the group members, because the leader has to continually reveal his identity and locations. Also, the use of the leader as a proxy for LBS access implies lack of end-to-end connectivity between the service provider and group members. Their scheme relies on public key cryptography in the PKI (public key infrastructure) setting and pseudonyms are required to achieve anonymity. However, no details are provided to manage these anonymous certificates for vehicles, which has been shown to be an obstacle to achieve practical anonymity in VANETs. Finally, their scheme does not provide anonymity revocability, which may not suit some applications in which anonymity must be revoked for the prevention, investigation, detection and prosecution of serious criminal offences.

In [14], the authors presented a secure data downloading (a specific LBS application) protocol in VANETs. This protocol overcomes the weaknesses in [12]. However, in [14], a single authority is employed to authenticate the vehicles and issue private/public key pairs for vehicles. Therefore, the system has the bottleneck of generating the key pairs for all the vehicles. Further, to download the data, the protocol requires five rounds of communication. Hence, it is not suited for real time applications. In [6], an anonymous batch authenticated and key agreement scheme for LBS in VANETs was presented. However, this scheme was shown to be insecure against a conspiracy attack [13].

[1] Privacy in VANETs usually denotes that an attacker cannot trace a vehicle according to the messages sent/received by a vehicle.

Contributions. We investigate the security and privacy concerns in LBSs for VANETs and propose a practical LBS scheme. For each LBS transaction, the scheme provides authentication, integrity and non-repudiation to both the service provider and the user. This guarantees the system's security. A user can obtain the service in an anonymous way and hence user privacy is well protected. However, if a user abuses the service, a tracing procedure can be invoked to find the malicious user, thereby efficiently preventing users from inappropriately leveraging the anonymity provided by the system. These two features are achieved with the help of group signatures. Our system does not depend on the grouping approach, which seems expensive in VANETs due to volatile connections. Instead, we propose that properly distributed RSUs in a VANET play a role similar to the one of the group leader in the AMOEBA protocol. This ensures that our system is robust in the sense that users can access LBSs and preserve their privacy without being affected by the vehicle density. To implement the scheme securely, we propose RSUs and the LBS provider to use their recognizable identities as their public keys. On the vehicle side, only a secret member key is needed to generate a group signature for authentication of data regarding the request of LBSs. This approach eliminates the certificate management overhead, noting that, as remarked above, neither identity based cryptosystems nor group signatures require public certificates. Analysis shows that cryptographic operations in our scheme introduce a very light overhead to the underlying VANET. Membership revocation is a critical issue in group signature based systems. The above system also suffers from the membership revocation problem. To handle this problem, we extend our system with hierarchical technology HKMK (see Section 6). With this technology, our system can also host a large number of LBS users.

2 System Architecture and Design Goals

2.1 System Architecture

The system architecture is illustrated in Figure 1. In the system, there are key generation center(s) (KGC(s)), RSUs, vehicles and providers of location based services (LBS providers):

- A KGC is a trusted third party. It generates private keys for vehicles and LBS providers and it issues secret member keys for vehicles. In addition, the KGC is assumed to be able to determine the real identity of vehicles and LBS providers.
- RSUs are equipped with on-board sensory, processing, and wireless communication modules, and they are distributed along the roadside. They are connected to LBS providers by a wired network. RSUs are assumed to be semi-trusted (*i.e.*, some of them might be compromised).
- Vehicles move along the roads, sharing environmental information with each other and/or querying LBSs through RSUs using the DSRC protocol [1]. Each vehicle is equipped with on-board sensory, processing, and wireless communication modules.
- LBS providers process the data forwarded by RSUs and offer LBSs to vehicles.

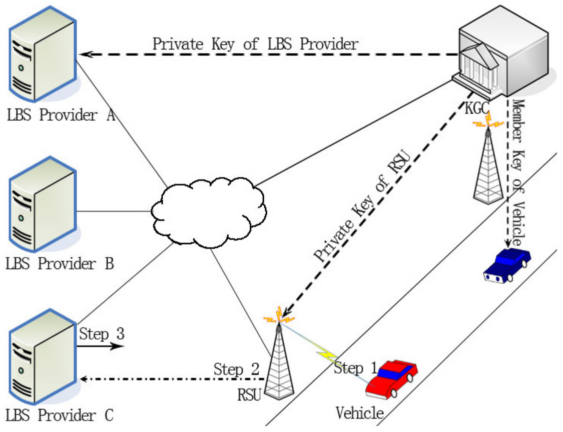

Fig. 1. System architecture

2.2 Design Goals

The design goals of our LBS scheme are summarized as follows.

- **Security.** Authentication, integrity and non-repudiation should be provided for both the service provider and the user. The transcripts of a successful execution of the LBS protocol can be used as evidence to prove that both parties have been involved in that transaction. If a user abuses the service, the real identity of the user can be traced.
- **Privacy.** An attacker cannot trace or profile any specific user by monitoring all the public communications in the system or data mining her received data.
- **Robustness.** Users should be able to access intended LBSs anytime anywhere, provided that the underlying VANET is available and the LBSs are open for providing services. The service quality and privacy of the requestor should not be affected by vehicle density.

We further decompose our design goals into security requirements. We first show the communication model of an LBS protocol. As illustrated in Figure 1, an LBS protocol can be described in three steps. In the first step, a vehicle sends its request to a nearby RSU in a single-hop or multi-hop manner. In the second step, the RSU receives the request from the vehicle and detects what kind of services the vehicle is asking for; then it forwards the request to the corresponding LBS provider. In the last step, the LBS provider authenticates the vehicle. If the vehicle is a subscriber, the LBS provider returns the requested service to the vehicle by routing his response through RSUs neighboring the RSU the vehicle request came from. Within the above model, the design goals can be achieved by meeting the following security requirements.

[Security requirements at Step 1] In this step the security requirements are:

– **Message Confidentiality.** An LBS request may contain sensitive information of a vehicle. For instance, if a vehicle requests a priced service, the e-cash may be included in the query. If the vehicle sends this request to the LBS provider through an RSU, anyone can learn the e-cash information and an attacker can steal the e-cash by disabling the communication from the requesting vehicle to the RSU. Hence, message confidentiality is required. Message confidentiality at this step has two levels. Level 1 provides confidentiality when the attacker does not know the private key of the RSU. It requires that only the vehicle and the RSU be aware of the information exchange. Level 2 provides confidentiality even if the attacker learns the private key of the RSU. In this case, we require that no one but the vehicle and the designated LBS provider can learn the content of the LBS request.
– **Vehicle Privacy.** Privacy in this step has also two levels. Level 1 guarantees privacy when the attacker does not know the private key of the RSU. It requires that it be computationally hard for everyone (except the message generator or some trusted third party) to decide whether two different messages were generated by the same vehicle. Level 2 guarantees privacy even when an RSU is compromised by an attacker. In this case, we require that the attacker can only learn the service type a vehicle is requesting.

[Security requirements at Step 2] In our system, an RSU serves as a router. For an LBS protocol in VANETs, an RSU only needs to know the service type that a vehicle is requesting so that it can forward the request to the right LBS provider. Hence, in our design, we only let RSUs learn the kind of service the vehicle wants to access. This step needs to meet the following security requirements:

– **Message Confidentiality.** No one but the vehicle and the designated LBS provider can learn the content of the message forwarded by the RSU.
– **Vehicle Privacy.** It is computationally hard for everyone (except the message generator or some trusted third party) to decide whether two different messages were generated by the same vehicle.

[Security requirements at Step 3] This step has the following requirements:

– **Vehicle Authentication.** The LBS provider must be sure that the request is from some registered vehicle, *i.e.*, a subscriber.
– **Vehicle Privacy.** The LBS provider only learns that a vehicle is querying the LBS but it cannot learn the vehicle's identity. Furthermore, the LBS provider cannot decide whether two different requests were generated by the same vehicle.
– **Vehicle Traceability.** In VANETs, the privacy of a vehicle should be conditional. That is, if necessary, some trusted third party should be able to revoke the anonymity of doubtable vehicles. Otherwise, a malicious vehicle might send fake messages to jeopardize the system without fear of being caught. In this paper, KGC is endowed with the ability to trace the real identity of dishonest vehicles sending fake messages to LBS providers in order to disrupt services.

3 Technical Preliminaries

3.1 Bilinear Maps

Bilinear maps are widely used in many cryptosystems, *e.g.*, identity-based cryptography (IBC) and group signature schemes. We briefly review them here.

Let $\mathbb{G}_1, \mathbb{G}_2$ be two cyclic groups of prime order q and $\mathbb{G}_T$ be a multiplicative cyclic group of the same order. Let g_1 denote a generator of $\mathbb{G}_1$, g_2 a generator of $\mathbb{G}_2$, ψ a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2) = g_1$. A mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ is called a bilinear mapping if $\hat{e}(g_1, g_2) \neq 1$ and $\hat{e}(g_1^{\alpha}, g_2^{\beta}) = \hat{e}(g_1, g_2)^{\alpha\beta}$ for all $\alpha, \beta \in Z_q^*$.

Such a bilinear map $\hat{e}$ can be constructed with the modified Weil pairing [3] on elliptic curves. ψ can be a trace map as described in [3], and when $\mathbb{G}_1 = \mathbb{G}_2$ and $g_1 = g_2$, ψ can be the identity map. For simplicity, in this paper we will take $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$ and $g_1 = g_2 = g$.

3.2 Identity-Based Cryptography

Identity-based cryptography (IBC) [2] was introduced to simplify certificate management procedures of public key infrastructures (PKI). In IBC, the public key of an entity is some unique public information about the identity of the entity (*e.g.*, the entity's location information). Therefore, the need for public key certification can be eliminated. The private key of an entity is generated by a trusted third party, Key Generation Center (KGC).

In VANETs, since we need not consider the privacy of RSUs and LBS providers, we can use the location information and service type as the identity of an RSU (and LBS provider). For instance, if an RSU is located at street A in city B, then we can use '$RSU, streetA, cityB$' as the identity of this RSU; for an LBS provider who provides online map service in city B, we can use '$onlinemap, cityB$' as the identity of the LBS provider.

3.3 Verifier-Local Revocation Group Signatures

Group signatures [5] allow the members of a group to sign on behalf of the group. Everyone can verify the signature with a group public key while no one can know the identity of the signer except the group manager. Further, except for the group manager, it is computationally hard for anyone to decide whether two different signatures were issued by the same group member.

Member revocation is needed to disable members who left the group or whose secret member key and/or member certificate were/was compromised. Most group signatures suffer from inefficient member revocation. Recently, an efficient approach to membership revocation in group signatures was proposed, called verifier-local revocation [4]. The idea is that only verifiers are involved in the revocation mechanism, while signers have no involvement. This approach is especially suitable for mobile environments where mobile signers (*i.e.,* the vehicles in our case) have much less computational power than the verifying servers (*i.e.*, the LBS providers).

In this paper, we use the optimized verifier-local revocation group signature scheme in [9] to achieve 'Vehicle Authentication', 'Vehicle Privacy' and 'Vehicle Traceability'. Pairing operation is the most time consuming operation in the scheme in [9]. By our optimization, to generate and verify a group signature, the number of pairing operations needs to compute can be reduced from 5 and 6 to 1 and 2 respectively.

4 A Privacy-preserving LBS Proposal

In this section, we propose a privacy-preserving LBS scheme in VANETs. We define the following notations to simplify the description. $\mathcal{V}$: a vehicle; $\mathcal{R}$: an RSU; $\mathcal{L}$: an LBS provider; $ID_{\mathcal{A}}$: the identity of entity $\mathcal{A}$; $S_{\mathcal{A}}$: the secret key of $\mathcal{A}$; $||$: message concatenation operation; Des: the description of an LBS request; Add: the addresses of some RSUs near $\mathcal{R}$ through which the LBS response to $\mathcal{V}$ will be routed; TP: a time stamp; IEK: the identity enrolment key, used to generate private keys for RSUs and LBS providers; MEK: the member enrolment key, used to issue member keys for vehicles; $\mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot)$: a symmetric key encryption scheme (*e.g.*, AES), with $\mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot)$ being the corresponding encryption/decryption algorithms, and K being a key which specifies the particular transformation of plaintext into ciphertext during encryption, or vice versa during decryption.

4.1 High Level Description

This section outlines the basic ideas of our LBS system for application in VANETs. We refer to the three steps mentioned in Section 2.2 and shown in Figure 2.

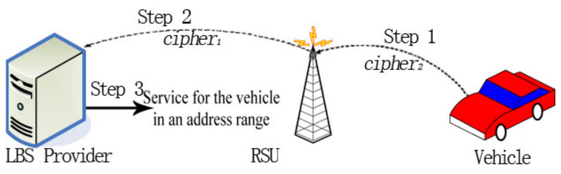

Fig. 2. The LBS protocol

In the first step, $\mathcal{V}$ first prepares a request of the form $Des||Add||TP||Sig$ and encrypts this request under $\mathcal{L}$'s identity to generate a ciphertext $cipher_1$, where Sig is the verifier-local revocation group signature on $Des||Add||TP$. Then it encrypts $TP||ID_{\mathcal{L}}||cipher_1$ under the identity of its nearby RSU to generate the ciphertext $cipher_2$. Finally, $cipher_2$ is sent to the RSU. In the second step, when the RSU receives $cipher_2$ from $\mathcal{V}$, RSU decrypts the ciphertext $cipher_2$ to get $TP||ID_{\mathcal{L}}||cipher_1$. If TP is fresh, RSU forwards $cipher_1$ to the LBS provider with identity $ID_{\mathcal{L}}$. In the last step, the LBS provider decrypts the ciphertext $cipher_1$ to get $Des||Add||TP||Sig$. It then checks whether TP is fresh and Sig is a valid signature on $Des||Add||TP$. If TP is fresh and Sig is valid, the LBS provider provides the corresponding service in Des to Add.

4.2 The Concrete Scheme

Our concrete LBS scheme consists of the following five stages.

[System Setup]

At this stage, KGC initializes the system-wide parameters as follows:

1. Choose a cyclic group $\mathbb{G}$ and a cyclic multiplicative group $\mathbb{G}_T$ of the same order q, so that there exists a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G}$ is generated by g; choose also $u_0, u_1 \in \mathbb{G}$.
2. Pick $\kappa, \rho \in Z_q^*$ as KGC's master secret key, and compute $u_2 = g^{\kappa}, u_3 = g^{\rho}$ as KGC's master public key. Hereafter, we will also denote κ as IEK and ρ as MEK.
3. Compute $Y_0 = \hat{e}(u_0, u_3), Y_1 = \hat{e}(u_0, g), Y_2 = \hat{e}(u_1, g), Y_3 = \hat{e}(g, g)$.
4. Choose a symmetric key encryption scheme $\mathcal{E}_K(.)/\mathcal{D}_K(.)$. We assume that the bit-length of K is λ.
5. Select cryptographic hash functions $H_0(\cdot) : \{0,1\}^* \longrightarrow \mathbb{G}$, $H_1(\cdot) : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$ and $H_2(\cdot) : \mathbb{G}_T \longrightarrow \{0,1\}^{\lambda}$.
6. Publish the system parameters as $\Psi = (\hat{e}, q, \mathbb{G}, \mathbb{G}_T, g, u_0, u_1, u_2, u_3, H_0, H_1, H_2, Y_0, Y_1, Y_2, Y_3, \mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot))$. Ψ is assumed to be pre-loaded in each vehicle, RSU and LBS provider.

The KGC also maintains a member list ML and a revocation list RL, where ML is kept secret while RL can be accessed by the LBS providers. We will define these lists later.

[Registration]

Before a vehicle, an LBS provider or an RSU joins a VANET, it registers with the KGC. The KGC generates a secret key or a member key for them using the following algorithms.

RSUJoin: This algorithm is used to generate the secret key for an RSU. Suppose that the identity of an RSU $\mathcal{R}_i$ is $ID_{\mathcal{R}_i}$. The KGC computes $S_{\mathcal{R}_i} = H_0(ID_{\mathcal{R}_i})^{\kappa}$.

ServiceJoin: This algorithm is used to generate the secret key for an LBS provider. Suppose that the identity of an LBS provider is $ID_{\mathcal{L}_i}$. The KGC computes $S_{\mathcal{L}_i} = H_0(ID_{\mathcal{L}_i})^{\kappa}$.

VehicleJoin: This algorithm is used to generate the member key for a vehicle. The KGC maintains a member list ML of tuples $(ID_{\mathcal{V}}, w, x, v)$, where $v = u_1^x$. When a vehicle wants to join the system, the KGC accepts a vehicle's identity $ID_{\mathcal{V}_i}$ and generates the member key as follows:

1. Select $x_i \in \mathbb{Z}_q^*$, compute $w_i = g^{1/(\rho+x_i)}$ and set (w_i, x_i) as $\mathcal{V}_i$'s member key.
2. Add $(ID_{\mathcal{V}_i}, w_i, x_i, v_i)$ to ML, where $v_i = u_1^{x_i}$ is the revocation token of $\mathcal{V}_i$.

[LBS Protocol]

As discussed in Section 2.2, an LBS protocol consists of three steps.

Step 1: The first step is the vehicle-to-RSU communication. Suppose that $\mathcal{V}_i$ wants to access the LBS provider $\mathcal{L}_k$ and its nearby RSU is $\mathcal{R}_j$. $\mathcal{V}_i$ chooses $s \in \mathbb{Z}_q^*$ and computes $C_1 = g^s$, $SK_1 = H_2(\hat{e}(u_2^s, H_0(ID_{\mathcal{L}_k})))$, where SK_1 will be the key of the symmetric key encryption scheme $\mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot)$, sets $m_1 = Des||Add||TP||Sig$, and computes $C_2 = \mathcal{E}_{SK_1}(m_1)$, where Sig is the signature on $m_0 = Des||Add||TP$ which is generated using the following SigGen algorithm:

1. Randomly select $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$ and compute $\delta = x_i\alpha, \zeta = x_i\beta, \eta = x_i\gamma$.
2. Compute $t_1 = w_i u_0^{\alpha}, t_2 = g^{\alpha} u_0^{\alpha}, T_3 = Y_2^{\eta}$ and $t_4 = g^{\gamma}$.
3. Select $r_\alpha, r_\beta, r_\gamma, r_{x_i}, r_\delta, r_\zeta, r_\eta \in \mathbb{Z}_q^*$ at random.
4. Compute $R_1 = g^{r_\alpha} u_0^{r_\beta}, R_2 = t_2^{r_{x_i}} g^{-r_\delta} u_0^{-r_\zeta}, R_3 = \hat{e}(t_1, g)^{-r_{x_i}} Y_0^{r_\alpha} Y_1^{r_\delta}, R_4 = Y_2^{r_\eta}, R_5 = g^{r_\gamma}, R_6 = t_4^{r_{x_i}} g^{-r_\eta}$.
5. Compute $c = H_1(\Psi, m_0, t_1, t_2, T_3, t_4, R_1, ..., R_6)$.
6. Compute $s_\alpha = r_\alpha + c\alpha, s_\beta = r_\beta + c\beta, s_\gamma = r_\gamma + c\gamma$, $s_{x_i} = r_{x_i} + cx_i, s_\delta = r_\delta - c\delta, s_\zeta = r_\zeta + c\zeta$ and $s_\eta = r_\eta + c\eta$.
7. Output the group signature $Sig = (t_1, t_2, T_3, t_4, c,\ s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$.

$\mathcal{V}_i$ chooses $t \in \mathbb{Z}_q^*$ and computes $C_3 = g^t$, $SK_2 = H_2(\hat{e}(u_2^t, H_0(ID_{\mathcal{R}_j})))$, sets $cipher_1 = C_1||C_2$, $m_2 = TP||ID_{\mathcal{L}_k}||cipher_1$, computes $C_4 = \mathcal{E}_{SK_2}(m_2)$, and sends $cipher_2 = (C_3||C_4)$ to $\mathcal{R}_j$.

Step 2: When $\mathcal{R}_j$ receives $cipher_2 = (C_3, C_4)$, it computes $SK_2 = H_2(\hat{e}(C_3, S_{\mathcal{R}_j}))$, $m_2 = TP||ID_{\mathcal{L}_k}||cipher_1 = \mathcal{D}_{SK_2}(C_4)$. It checks TP to decide whether the request is fresh. If it is, it sends $cipher_1 = (C_1, C_2)$ to $\mathcal{L}_k$; otherwise it aborts.

Step 3: When $\mathcal{L}_k$ receives $cipher_1 = (C_1, C_2)$, it computes $SK_1 = H_2(\hat{e}(C_1, S_{\mathcal{L}_k}))$, $m_1 = Des||Add||TP||Sig = \mathcal{D}_{SK_1}(C_2)$. If TP is not fresh, it aborts. Otherwise, it extracts $Sig = (t_1, t_2, T_3, t_4, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$, and checks whether Sig is a valid group signature on $Des||Add||TP$ using the following SigVer algorithm:

1. Compute $R_1 = g^{s_\alpha} u_0^{s_\beta} t_2^{-c}, R_2 = t_2^{s_{x_i}} g^{-s_\delta} u_0^{-s_\zeta}, R_3 = \hat{e}(t_1, g^{-s_{x_i}} + u_3^{-c}) Y_0^{s_\alpha} Y_1^{r_\delta} Y_3^{c}, R_4 = Y_2^{s_\eta} T_3^{-c}, R_5 = g^{s_\gamma} t_4^{-c}, R_6 = t_4^{s_{x_i}} g^{-s_\eta}$.
2. Check $c \overset{?}{=} H_1(\Psi, m_0, t_1, t_2, T_3, t_4, R_1, ..., R_6)$.
3. Output *valid* if the signature passes the above check.

If the signature is valid, $\mathcal{L}_k$ provides the service to *Add* according to *Des*[2].

[2] In *Des*, an AES key and an identifier (e.g., a random number) can be included. The outcome could be encrypted by the LBS provider under that AES key and broadcasted with the identifier, so that only the vehicle who generated the request needs to decrypt the encrypted outcome and any other vehicles can not read the content of the outcome.

[Revocation]

Two mechanisms are suggested to tackle the revocation problem. Firstly, the KGC maintains a revocation list RL. Under normal circumstances, when a vehicle $\mathcal{V}_i$ is compromised, the KGC first finds the corresponding $(ID_{\mathcal{V}_i}, w_i, x_i, v_i)$ in ML and then adds v_i to the revocation list RL. To detect whether a group signature $Sig = (t_1, t_2, T_3, t_4, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$ is generated by a revoked vehicle, the LBS provider checks $T_3 \overset{?}{=} e(t_4, v_j)$ for all $v_j \in \mathsf{RL}$. If none of the equations holds, it means that the vehicle is not revoked. Secondly, when there are too many revoked vehicles in RL, we may allow the KGC to choose a threshold τ; and when the number of revoked vehicles in RL is greater than τ, the KGC updates its MEK and corresponding public key, and re-issues member keys for all the vehicles. This mechanism gives a tradeoff between revocation checks by LBS providers and key updates for entities in a VANET. The key updates may cause heavy overhead in case of a very large-scale VANET. In Section 6, we further propose a hierarchical approach to alleviate the overhead so that the system can stay efficient even if the VANET hosts a large number of vehicles.

[Trace]

Let $Sig = (t_1, t_2, T_3, t_4, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$ be a valid group signature. To trace a vehicle, the KGC checks $T_3 \overset{?}{=} e(t_4, v_i)$ for the tuple $(ID_{\mathcal{V}_i}, w_i, x_i, v_i)$ on ML. If this equation holds, KGC outputs $ID_{\mathcal{V}_i}$.

5 Evaluation

5.1 Security Analysis

The following analysis shows that the proposal meets all the security requirements described in Section 2.2.

Firstly, we show that the message confidentiality and the vehicle privacy of Step 1 are satisfied.

- *Level 1 message confidentiality and vehicle privacy.* At this step, the ciphertext $cipher_2$ is generated by using the basic identity-based encryption (IBE) scheme which was proven secure by Boneh and Franklin [2]. Therefore, the level 1 message confidentiality naturally follows. Furthermore, in each session of the protocol, a random value t is chosen. Therefore, in each session, C_3 and C_4 are different and independent from those in other sessions.
- *Level 2 message confidentiality and vehicle privacy.* In our protocol, the content of the LBS request is encrypted under the LBS provider's identity in $cipher_1$ using the Boneh and Franklin IBE scheme [2]. Only the designated LBS provider owns the secret key corresponding to this identity. Hence, even if the private key of the RSU is leaked to the attacker, no one except the vehicle and the designated LBS provider can read the content of the LBS request. Furthermore, in each session of the protocol, a random value s is chosen. Therefore, in each session, C_1 and C_2 are also different and independent from those in other sessions.

In Step 2, the RSU can decrypt $cipher_2$ to get $TP||ID_{\mathcal{L}_k}||cipher_1$. From $ID_{\mathcal{L}_k}$, the RSU can learn what kind of service the vehicle wants to access. However, since the RSU does not know the private key of the LBS provider, it cannot learn the content of $cipher_1$. Therefore, 'Message Confidentiality' for this step is met. Furthermore, $cipher_1$ is also generated under the Boneh and Franklin IBE scheme. The 'Vehicle Privacy' for this step accordingly follows.

Finally, we show that our protocol meets 'Vehicle Authentication', 'Vehicle Privacy' and 'Vehicle Traceability' of Step 3 defined in Section 2.2. In this step, the LBS provider first decrypts the ciphertext $cipher_1$ to get $Des||Add||TP||Sig$. If Sig is a valid group signature, then the LBS provider is sure that the request comes from a registered vehicle. Hence, 'Vehicle Authentication' is satisfied. Furthermore, the KGC can recover the identity of the vehicle, so 'Vehicle Traceability' is satisfied. As to 'Vehicle Privacy', since anyone can generate $Des||Add||TP$, it is easy to see that $Des||Add||TP$ may not help the LBS provider to trace a vehicle. It remains Sig for the LBS provider to trace a vehicle. However, the verifier-local revocation group signature has the property that it is computationally hard for anyone but the trusted third party (KGC in our scheme) to decide whether two different signatures were issued by the same member. Hence, Sig cannot help the LBS provider to trace a vehicle.

5.2 Transmission Overhead

In this section, we examine the transmission delay incurred by the security and privacy mechanism. We will only deal with the delay in Step 1, which has a relatively crucial bandwidth limitation.

From our LBS protocol, it is easy to see that the length of an LBS request is equal to the length of $C_3||TP||ID_{\mathcal{L}}||C_1||Des||Add||TP||Sig$ in Step 1. Excluding $Des||Add$[3], it remains to see the length of $C_3||TP||ID_{\mathcal{L}}||C_1||TP||Sig$. According to [3] and [9], the length of a point in $\mathbb{G}$ and the length of Sig are 171 bits (about 22 bytes) and 362 bytes respectively. In addition, the length of TP is 4 bytes and the length of $ID_{\mathcal{L}}$ is 20 bytes. Hence, the length of $C_3||TP||ID_{\mathcal{L}}||C_1||TP||Sig$ is about 434 bytes. According to DSRC [1], the minimal data rate in DSRC is 6 Mbps. Hence, we have that the maximal transmission delay caused by security and privacy mechanism at Step 1 is $\frac{434\times 8}{6\times 1024\times 1024}$ s ≈ 0.55 ms. This delay is very low for vehicles in VANETs.

5.3 Computational Overhead

This section discusses the computational overhead at each step in our protocol. In the sequel, we will only consider the costly operations (*i.e.* pairing and point exponentiation operations). According to the execution time results shown in [7], the measured processing time for one bilinear pairing operation is about 1.87 ms and the time for one point exponentiation operation is about 0.49 ms.

[3] These data are required even without any security and privacy mechanism. It is clearer to see the cryptographic overhead without considering these data.

In the first step, we notice that all point exponentiation operations can be pre-computed off-line. Therefore, this step only needs to compute two pairing operations on-line. The time is about 3.74 ms. In the second step, an RSU only needs to compute one pairing operation to decrypt the ciphertext $cipher_2$. The time is about 1.87 ms. For the last step, the LBS provider needs to calculate 17 point exponentiation operations and 2 pairing operations. The total time is about 12.07 ms. Therefore, the total computational overhead for all the steps is about 17.68 ms. This is affordable for vehicles wishing to access LBS.

6 Hierarchical KGC and Multi-issue Key

In our basic system, we use a single KGC to generate private keys for RSUs and LBS providers, and issue member keys for vehicles. However, if there is a huge number of users in a VANET, the KGC may become a bottleneck: the KGC needs not only to generate private keys or member keys for a large number of users, but also to verify the identities of the users. Furthermore, as the number of vehicles in the revocation list grows, the performance of the system might decline. To let the system remain efficient even if a large number of vehicles are revoked, we introduce an approach referred to as hierarchical KGCs and multi-issue key (HKMK). The idea of HKMK is illustrated in Figure 3.

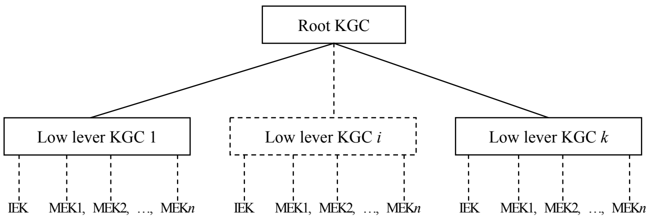

Fig. 3. Hierarchical KGC and multi-issue key

In this approach, we use a two-level hierarchical KGC. A root KGC is used to issue certificates for low-level KGCs. As in our basic system, each low-level KGC has a single identity enrolment key (IEK) which is used to generate private keys for RSUs and LBS providers. However, unlike in our basic system, each low-level KGC has n different member enrolment keys (MEKs). When a vehicle joins the system, the low-level KGC randomly chooses one of its MEKs and generates a member key for this vehicle. In this way, vehicles in a domain are separated into n sub-groups and, when a vehicle contacts an LBS provider for the LBS, the LBS provider can only learn that the vehicle belongs to a sub-group.

In what follows, we show how to set up the system parameters in the *System Setup* stage. We reformulate this stage into two sub-stages: *Root KGC Setup* and *Low-Level KGC Setup*. The description of each sub-stage comes as follows.

[Root KGC Setup]

The root KGC initializes the system-wide parameters as follows:

1. Choose $\hat{e}, \mathbb{G}, \mathbb{G}_T, q, \mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot), H_0, H_1, H_2$ as those in the System Setup stage in Section 4.2.
2. Choose $u_0, u_1 \in \mathbb{G}$ and compute $Y_1 = \hat{e}(u_0, g), Y_2 = \hat{e}(u_1, g), Y_3 = \hat{e}(g, g)$.
3. Publish the system-wide parameters $\Psi = (\hat{e}, q, \mathbb{G}, \mathbb{G}_T, g, u_0, u_1, H_0, H_1, H_2, Y_1, Y_2, Y_3, \mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot))$. Ψ is assumed to be pre-loaded in each low-level KGC, vehicle, RSU and LBS provider.

[Low-Level KGC Setup]

After seeing the system-wide parameters, a low-level KGC generates its own parameters as follows:

1. Pick $\kappa \in Z_q^*$ as its identity enrolment key (IEK) and n member enrolment keys (MEKs) $\rho_1, ..., \rho_n \in Z_q^*$.
2. Compute $u_2 = g^{\kappa}$ and $u_{3i} = g^{\rho_i}, 1 \leq i \leq n$ as its master public key.
3. Compute $Y_{0i} = \hat{e}(u_0, u_{3i}), 1 \leq i \leq n$.
4. Publish the parameters as $\Omega = (u_2, u_{31}, ..., u_{3n}, Y_{01}, ..., Y_{0n})$.

The low-level KGC also maintains n member lists $\mathsf{ML}_1, ..., \mathsf{ML}_n$ and revocation lists $\mathsf{RL}_1, ..., \mathsf{RL}_n$ corresponding to the n MEKs, respectively. To deal with the revocation problem more efficiently, similarly to our basic system, a low-level KGC chooses a threshold τ. If a vehicle $\mathcal{V}_i$ is compromised (we assume the member key of $\mathcal{V}_i$ is issued by using the j-th MEK, $1 \leq j \leq n$) and the number of vehicles in RL_j is not greater than τ, the low-level KGC first finds the revocation token of $\mathcal{V}_i$ in ML_j, then adds the revocation token to RL_j. Otherwise, the KGC updates its j-th MEK and corresponding public key u_{3j} and $Y_{0j} = \hat{e}(u_0, u_{3j})$, and re-issues member keys for all the vehicles in j-th sub-group.

Acknowledgments and Disclaimer. This work was supported in part by the the NSF of China under Grants 61202465, 61021004, 11061130539, 60970114, 60970115, 91018008, 60970116, 61173154, 61003214, 61173192, 61103222; EU FP7 under Projects "DwB" and "Inter-Trust"; the Spanish Government under Projects CTV-09-634, PTA2009-2738-E, TSI-020302-2010-153, TIN2009-11689, TIN2011-27076-C03-01, CONSOLIDER INGENIO 2010 "ARES" CSD2007-0004, and TSI2007-65406-C03-01; the Government of Catalonia under Grant SGR2009-1135; the Fundamental Research Funds for the Central Universities of China under Project 3103004; the Open Project of Shanghai Key Laboratory of Trustworthy Computing (No. 07dz22304201101); the Shanghai NSF under Grant No. 12ZR1443500, 11ZR1411200; the Shaanxi Provincial Education Department under Scientific Research Program 2010JK727 and the Shanghai Chen Guang Program (12CG24). J. Domingo-Ferrer was supported in part as an ICREA-Acadèmia researcher by the Catalan Government.

References

1. Dedicated Short Range Communications (DRSC) home, `http://www.leearmstrong.com/Dsrc/DSRCHomeset.htm`
2. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
3. Boneh, D., Lynn, B., Shacham, H.: Short Signatures from the Weil Pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
4. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: ACM Conference on Computer and Communications Security, pp. 168–177 (2004)
5. Chaum, D., van Heyst, E.: Group Signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
6. Huang, J., Yeh, L., Chien, H.: 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)
7. Jiang, Y., Shi, M., Shen, X., Lin, C.: BAT: A robust signature scheme for vehicular networks using binary authentication trees. IEEE Transactions on Wireless Communications 8(4), 1974–1983 (2009)
8. Kocher, P.C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
9. Nakanishi, T., Funabiki, N.: Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 533–548. Springer, Heidelberg (2005)
10. Papadimitratos, P., Gligor, V., Hubaux, J.: Securing vehicular communications - Assumptions, requirements, and principles. In: Workshop on Embedded Security in Cars, ESCAR 2006 (2006)
11. Raya, M., Hubaux, J.: Securing vehicular ad hoc networks. Journal of Computer Security 15(1), 39–68 (2007)
12. Sampigethaya, K., Li, M., Huang, L., Poovendran, R.: AMOEBA: Robust location privacy scheme for VANET. IEEE Journal on Selected Areas in Communications 25(8), 1569–1589 (2007)
13. Wang, H., Zhang, Y.: On the security of an anonymous batch authenticated and key agreement scheme for value-added services in VANETs. Procedia Engineering 29, 1735–1739 (2012)
14. Yong, H., Jin, T., Yu, C., Chi, Z.: Secure data downloading with privacy preservation in vehicular ad hoc networks. In: IEEE International Conference on Communications 2010, pp. 1–5 (2010)
15. 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)
16. 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)