Practical Secure and Privacy-Preserving Scheme for Value-Added Applications in VANETs

Lei Zhang[a,e,*], Qianhong Wu[b,f], Bo Qin[c,g], Josep Domingo-Ferrer[d], Bao Liu[a]

[a] *Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, China, China*
[b] *School of Electronics and Information Engineering, Beihang University, China*
[c] *Key Laboratory of Data Engineering and Knowledge Engineering, School of Information, Ministry of Education, Renmin University of China, Beijing, China*
[d] *UNESCO Chair in Data Privacy, Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Tarragona, Catalonia*
[e] *State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China*
[f] *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[g] *Key Laboratory of Cryptologic Technology and Information Security, State Key Laboratory of Cryptology, Ministry of Education, Shandong University, Beijing, China*

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 secure and privacy-preserving scheme for value-added applications in VANETs; specifically, a vehicular secure and privacy-preserving location-based service (LBS). For each LBS transaction, the scheme provides authentication, integrity and non-repudiation for both the service provider and the vehicle. A vehicle can obtain the service in an anonymous way and hence vehicle privacy is well protected. However, if a vehicle maliciously uses the service, a tracing procedure can be invoked to find the malicious vehicle, thereby efficiently preventing vehicles from abusing the anonymity provided by the system. These features are achieved by efficiently exploiting the

*Corresponding author.
Email addresses: leizhang@sei.ecnu.edu.cn (Lei Zhang), qianhong.wu@buaa.edu.cn (Qianhong Wu), bo.qin@ruc.edu.cn (Bo Qin), josep.domingo@urv.cat (Josep Domingo-Ferrer), baoliu@ecnu.edu.cn (Bao Liu)

available infrastructure of VANETs and without requiring cooperation from other vehicles, which might be reluctant to cooperate or otherwise unavailable. Although public-key cryptosystems are employed as building blocks, no complicated certificate management is required by our system.

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

---

1. Introduction

Vehicular *ad hoc* networks (VANETs), consisting of computers mounted on vehicles and road infrastructures, are emerging as the first commercial mobile *ad hoc* networks. This kind of networks allows 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.* [18, 21, 24, 26, 34]) has appeared to guarantee trustworthiness of vehicle-generated messages and address the VANET security concerns.

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 [1]. 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 ought to 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; examples of such technologies are anonymous credential systems, First Virtual, SSL, iKP secure electronic payments [6] and the SET protocol [5]. However, VANETs are very dynamic and their communications are volatile, which makes those complex protocols unsuitable.

[1] Privacy in VANETs usually means that an attacker cannot trace a vehicle by using the messages the latter sends/receives.

1.1. Related Work

In recent years, a number of papers have dealt with security and privacy in VANETs. Various security and privacy challenges in vehicular networks are discussed in [13, 18, 24, 25, 26, 34]. To address these challenges, two techniques are generally used. The first is based on anonymous certificates and the second on group signatures.

The seminal proposals using the first technique are due to Raya and Hubaux [27, 28]. In their proposals, digital signatures are combined with short-lived anonymous certificates to meet the security requirements of the vehicular nodes. However, using their approach, each vehicle needs to preload a huge pool of anonymous certificates, and the trusted authority also needs to maintain all the anonymous certificates of all vehicles, thus incurring a heavy certificate management overhead. Later, Zhang *et al.* [35] designed an efficient conditional privacy-preserving protocol for vehicular communications (where "conditional" means that the privacy of dishonest users can be revoked) using identity-based cryptography [30] to avoid complicated certificate management. However, their approach relies on idealized tamper-proof devices embedded in the vehicles. The main aim of the tamper-proof devices is to generate random pseudo-identities, which are used similarly to anonymous certificates. For the protocol to operate properly, the private key of the trusted authority should be stored in the tamper-proof devices. However, it cannot be ignored that an adversary might gain access to those devices by some powerful attacks, *e.g.*, side-channel attacks [19, 31].

Following the group signature based research line, Guo *et al.* [15] presented a novel security framework for vehicular communications based on group signatures. However, neither a concrete instantiation nor an experimental analysis are given in [15]. The first concrete instantiation of a group signature based technique in VANETs is due to Lin *et al.* [20], who presented a conditional privacy-preserving vehicular communications protocol. More recently, additional efforts have been made to improve the trustworthiness of vehicle-generated messages [36] or to achieve robustness and scalability in VANETs [37]. The main advantage of group signature based schemes over the pseudonym based ones is that the former do not need to manage a large number of anonymous certificates.

Compared with the extensive attention received by safety applications of VANETs, few efforts have been made to address the security and privacy issues of value-added applications luring drivers into using VANETs. To fill this gap, Sampigethaya *et al.* [29] 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. Within their group concept, a group leader (a single vehicle in the group) is selected to represent the group, and used as a proxy for LBS access. That is, the group leader submits LBS queries on behalf of the group members and, hence, the privacy of members other than the group leader is naturally achieved. However, as remarked by the authors of [29], there are still several loose ends in their scheme and further work is needed to enable it to be widely deployed in the real world. 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 remains unchanged, the privacy of group members comes at the cost of sacrificing the privacy of the group leader, who must continually reveal its 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 on how to manage these anonymous certificates for vehicles, which has been shown to be an obstacle to achieve practical anonymity in VANETs. Their scheme also requires an anonymous electronic payment as a building block to complete a privacy-preserving LBS transaction, but no details are given. Indeed, there are some subtleties to implement an anonymous e-payment protocol in their scheme. For instance, if the group leader as a proxy requests service for a group member and then receives the answer from the LBS provider, the group member might not be able to obtain the answer if the member or the leader have left the group in the interim; note that the group is formed in an *ad hoc* and very volatile way. Then a dispute would arise as to whether the user should pay for a service given by the LBS provider but not received by the user. This is especially the case if payment is based on a successful transaction. Finally, the scheme [29] does not provide anonymity revocability, which may be a problem in applications in which anonymity must be revocable to prevent, investigate, detect and prosecute serious criminal offences [2].

In [33], the authors presented a protocol for secure data downloading (a specific LBS application) in VANETs. This protocol overcomes the weaknesses of [29]. However, in [33], a single authority is employed to authenticate the vehicles and issue private/public key pairs for them. 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, this protocol is not suited for real-time applications. In [17], an anonymous batch-authenticated and key agreement scheme for LBS in VANETs is presented. However, it was shown in [32] that the scheme in

[17] is insecure against a conspiracy attack.

1.2. Contributions and Paper Organization

In the earlier version [39] of this paper, we investigated the security and privacy concerns in LBSs for VANETs and proposed a practical LBS scheme. For each LBS transaction, the scheme provides authentication, integrity and non-repudiation for both the service provider and the vehicle. This guarantees security. A vehicle can obtain the service in an anonymous way and hence vehicle privacy is guaranteed too. However, if a vehicle abuses the service, a tracing procedure can be invoked to find the malicious user, thereby efficiently preventing vehicles from illegitimately leveraging anonymity. Our system does not depend on the group concept (which assumes that a vehicle is selected to represent the group) used in [29] to protect the privacy of the group members; this is good, because we argued above that volatile connections in VANETs make the lifetime of groups very short. Instead, we employ group signatures to protect the privacy of vehicles and also allow any vehicle to submit an LBS request by itself. Further, we propose the (properly distributed) RSUs in the VANET to play a similar role as the group leader in the AMOEBA protocol. This ensures that our system is robust in the sense that vehicles can access LBSs and preserve their privacy without being affected by the vehicle density. To implement the scheme securely, we propose the RSUs and the LBS provider to use their recognizable identities as their public keys. On the vehicle's side, only a secret member key is needed to generate a group signature to authenticate data in the LBS request. This approach eliminates the certificate management overhead, because, as remarked above, neither identity-based cryptosystems nor group signatures require public-key certificates.

The above LBS scheme is based on symmetric pairings (see Section 3.1). However, symmetric pairings are quite restrictive in what regards the choice of curves. Asymmetric pairings have the following advantages over symmetric pairings: they are less special, they have better security properties and they yield substantially shorter signatures. In this version, we enhance the scheme in [39] by using asymmetric pairings. Our analysis shows that cryptographic operations in our LBS scheme are reasonable on the underlying VANET.

Membership revocation is a critical issue in group signature based systems. The above system also suffers from the membership revocation problem. As the number of revoked vehicles in the system grows, the performance of the system declines. To handle this problem, in the earlier version [39] of this paper, we equipped our system with the HKMK hierarchical technology

(which stands for Hierarchical Key-generation centers and Multi-issue Keys). With this technology, our system can host a large number of LBS users. Further, even if a large number of vehicles is revoked, our system remains efficient. However, in most systems based on group signatures (including HKMK), if a vehicle is revoked, an LBS provider can link the previous LBS requests of the vehicle. Thus, the privacy of the revoked vehicle is violated. To alleviate this problem, we propose the extended HKMK technology. In extended HKMK, the concept of time intervals [23] is adopted. Each vehicle has T revocation tokens corresponding to T time intervals, respectively. When a vehicle is revoked at interval i, signatures generated by the vehicle before interval i remain unlinkable. Therefore, extended HKMK effectively protects the privacy of the vehicles in the system.

The rest of the paper is organized in the following way. In Section 2, we describe our system architecture and design goals. Technical preliminaries are given in Section 3. Section 4 proposes our basic LBS scheme. We evaluate the new protocol in Section 5. Section 6 develops improved LBS systems. Finally, Section 7 concludes the paper.

2. System Architecture and Design Goals

2.1. System Architecture

The system architecture is illustrated in Figure 1. It consists of a key generation center (KGC), distributed RSUs, moving vehicles and a variety of 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.

- An RSU is equipped with sensory, processing, and wireless communication modules, and they are distributed along the roadside. The RSUs are connected to LBS providers by a wired network. RSUs are assumed to be semi-trusted (*i.e.*, some of them might be compromised).

- A vehicle moves along the roads, sharing environmental information with other vehicles nearby and/or querying LBSs through RSUs using the DSRC protocol [3]. For this purpose, the vehicle is equipped with on-board processing units (OBUs), wireless transmitter, and sensing device. Unlike the nodes in most mobile *ad hoc* networks, the vehicles

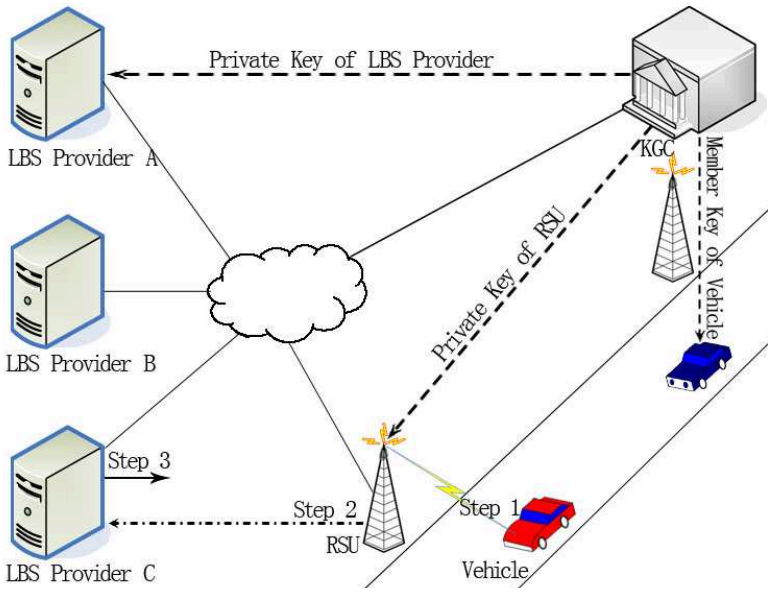

Figure 1: System architecture

in a VANET are assumed to be computationally powerful (OBUs are like personal computers). It is assumed that the vehicle has ample power supply (a car has powerful batteries and it can keep on recharging its OBUs as it runs).

- An LBS provider processes the LBS request (of the vehicles) forwarded by the RSUs and offers the corresponding services to the requesting vehicles.

2.2. Security and Privacy Requirements

The security and privacy requirements of our LBS scheme are summarized as follows.

- **Message Confidentiality.** Whenever the content of a message is sensitive, it should stay secret to anyone except the message sender and the legitimate receiver. Indeed, an LBS request may contain sensitive information of a vehicle. For instance, if a vehicle requests a priced service, the e-cash (usually a blind signature [11]) may be included in the query; if confidentiality was not satisfied, anyone could learn the e-cash information and steal it.

- **Vehicle Authentication.** The LBS provider ought to be sure that the request comes from a registered vehicle, *i.e.*, a subscriber.

- **Vehicle Privacy.** Unlinkable anonymity ought to be provided for the vehicle originating a message. That is, neither the LBS provider nor an attacker ought to be able to learn the identity of the vehicle having originated a message; furthermore, it ought to 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. In this way, a specific vehicle cannot be traced/profiled by monitoring/mining all messages it generates. Note that message confidentiality as described above usually implies vehicle privacy; however, we require message confidentiality only for sensitive content, whereas we require vehicle privacy for any message; for this reason, we consider vehicle privacy separately from message confidentiality.

- **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.

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 using the DSRC protocol 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 its response through RSUs neighboring the RSU the vehicle request came from. We next examine the security and privacy requirements to be satisfied at each step.

The attackers in our protocol can be classified into two types: external attackers who do not know the private key of RSUs or LBS providers and internal attackers, i.e., a routing RSU or an LBS provider. The first step should satisfy message confidentiality and vehicle privacy. In our scheme, we assume that an RSU can be compromised. Message confidentiality and vehicle privacy in this step can be provided at two levels. Level 1 provides

confidentiality and privacy against external attackers. Level 2 provides confidentiality and privacy even if an attacker learns the private key of the routing RSU (or the attacker is a malicious RSU); in this case, we require that the (inside) attacker can only learn the service type that a vehicle is requesting, but no one (except the vehicle and the designated LBS provider) should be able to learn the content of the LBS request. In the second step, 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 type of service the vehicle wants to access. This step needs to satisfy message confidentiality and vehicle privacy against external attackers and malicious RSUs. For the last step, firstly an LBS provider must be sure that the request comes from some registered vehicle. Hence, this step must satisfy vehicle authentication. Secondly, it requires that the LBS provider can only learn that a vehicle is querying the LBS but it cannot learn the vehicle's identity and cannot decide whether two different requests were generated by the same vehicle. In other words, vehicle privacy should be guaranteed even the LBS provider is malicious. Thirdly, 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 and jeopardize the system without fear of being caught. Hence, this step should satisfy vehicle traceability. Finally, the LBS provider should only return the requested service to the authenticated vehicles. This step also needs to consider message confidentiality.

3. Technical Preliminaries

3.1. Bilinear Maps

Bilinear maps are widely used in many cryptosystems, *e.g.*, identity-based cryptography (IBC) [38] and group signature schemes [23]. 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 map if it satisfies the following properties:

1. *Bilinearity.* $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ for all $a, b \in Z_q^*$.
2. *Non-degeneracy.* $\hat{e}(g_1, g_2) \neq 1$.

3. *Computability.* There exists an efficient algorithm to compute $\hat{e}(u, v)$ for any $u \in \mathbb{G}_1, v \in \mathbb{G}_2$.

Such a bilinear map $\hat{e}$ can be constructed with the modified Weil [22], Tate [14] or Eta [16] pairings on elliptic curves. ψ can be a trace map as described in [9] and when $\mathbb{G}_1 = \mathbb{G}_2$ and $g_1 = g_2$, ψ can be the identity map. In this paper, we say the pairing is asymmetric if $\mathbb{G}_1 \neq \mathbb{G}_2$; otherwise, the pairing is symmetric. Symmetric pairings are quite restricted in terms of the choice of curves. Asymmetric pairings have the advantage of being less special than symmetric ones and they have better security properties. Furthermore, they can lead to considerably shorter signatures than symmetric pairings.

3.2. Identity-Based Cryptography

Identity-based cryptography (IBC) was introduced by Shamir [30] 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. A trusted third party, called the Key Generation Center (KGC), generates the corresponding private keys for the entities in IBC.

To operate, the KGC first publishes the system parameters and keeps secret the corresponding master key. Given the system parameters, any party can compute the public key corresponding to an identity ID by combining the system parameters with the identity value. To obtain a corresponding private key, the party authorized to use the identity ID contacts the KGC, and the KGC uses the master key to generate the private key for the identity ID.

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, street A, city B'* as the identity of this RSU; for an LBS provider who provides online map service in city B, we can use *'online map, city B'* as the identity of the LBS provider.

3.3. Verifier-Local Revocation Group Signatures

Group signatures [12] allow the members of a group to sign on behalf of the group. If a signature is valid under a group public key, the signature verifier is sure that the signature was generated by a member of the group. This property can be used to achieve vehicle authentication in our scheme.

A group signature also has the property that, except for the group manager, it is computationally hard for anyone to decide whether two different signatures were issued by the same group member. This property satisfies the vehicle privacy requirement. Further, the group manager can recover the real identity of a signature generator. Therefore, the privacy of a group signature is conditional. Thus, the requirement of vehicle traceability is also met by using group signatures.

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 [10]. 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 [23] to achieve vehicle authentication, vehicle privacy and vehicle traceability. The verifier-local revocation group signature scheme in [23] is implemented with symmetric pairing. As mentioned in Section 3.1, asymmetric pairings have better security properties and can lead to considerably shorter signatures than symmetric pairings. To enjoy the advantages of asymmetric pairings, we convert the verifier-local revocation group signature scheme in [23] into one based on asymmetric pairings. We also optimize the signature generation and verification algorithms in [23]. Through our optimization, the number of pairing operations needed to generate and verify a group signature can be reduced from 5 and 6 to 1 and 2, respectively. Since the pairing operation is the most time-consuming one in the scheme of [23], the optimized verifier-local revocation group signature scheme is much more efficient than the original one.

4. Privacy-preserving LBS Proposal

In this section, we propose a secure and privacy-preserving LBS scheme in VANETs. Before describing the scheme in detail, we first list in Table 1 the notations used.

Table 1: Description of notation

Notation	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), such that $\mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot)$ is the corresponding encryption/decryption algorithm, where K is a key which specifies the particular transformation of plaintext into ciphertext during encryption, or vice versa during decryption.

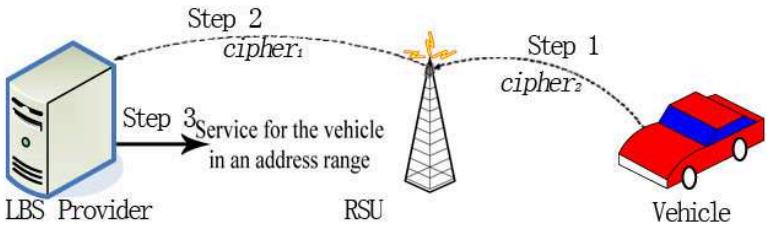

Figure 2: The LBS protocol

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 described in Section 2.2 and shown in Figure 2.

In the first step, the vehicle $\mathcal{V}$ first prepares a request of the form $Des||Add||TP||Sig$ and encrypts this request under the LBS provider $\mathcal{L}$'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 $cipher_2$ 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

In this section, we propose our concrete LBS scheme, consisting of the following five stages.

[System Setup]

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

1. Choose three cyclic multiplicative groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of the same order q, so that there exists a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $\mathbb{G}_1$ is generated by g_1 and $\mathbb{G}_2$ is generated by g_2.
2. Choose $u_0 \in \mathbb{G}_1$ and $u_1 \in \mathbb{G}_2$.

3. Randomly pick $\kappa, \rho \in Z_q^*$ as KGC's master secret key, and compute $u_2 = g_1^{\kappa}, u_3 = g_2^{\rho}$ as KGC's master public key. Hereafter, the names IEK and MEK will also be used for κ and ρ, respectively.
4. Compute $Y_0 = \hat{e}(u_0, u_3), Y_1 = \hat{e}(u_0, g_2), Y_2 = \hat{e}(g_1, u_1), Y_3 = \hat{e}(g_1, g_2)$.
5. Choose a symmetric key encryption scheme $\mathcal{E}_K(.)/\mathcal{D}_K(.)$. We assume that the bit-length of K is λ.
6. Select cryptographic hash functions $H_0(\cdot) : \{0,1\}^* \longrightarrow \mathbb{G}_2$, $H_1(\cdot) : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$ and $H_2(\cdot) : \mathbb{G}_T \longrightarrow \{0,1\}^{\lambda}$.
7. Publish the system parameters as $\Psi = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, u_0, u_1, u_2, u_3, H_0, H_1, H_2, \mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot), Y_0, Y_1, Y_2, Y_3)$. 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. Randomly select $x_i \in \mathbb{Z}_q^*$.
2. Compute $w_i = g_1^{1/(\rho + x_i)}$ and set (w_i, x_i) as the member key of $\mathcal{V}_i$.
3. 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 $\mathcal{R}_j$. $\mathcal{V}_i$ does the following:

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 $t_1 = w_i u_0^{\alpha}, t_2 = g_1^{\alpha} u_0^{\beta}, T_3 = Y_2^{\eta}$ and $t_4 = g_1^{\gamma}$.
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

$$\begin{cases} R_1 = g_1^{r_\alpha} u_0^{r_\beta} \\ R_2 = t_2^{r_{x_i}} g_1^{-r_\delta} u_0^{-r_\zeta} \\ R_3 = \hat{e}(t_1, g_2)^{-r_{x_i}} Y_0^{r_\alpha} Y_1^{r_\delta} \\ R_4 = Y_2^{r_\eta} \\ R_5 = g_1^{r_\gamma} \\ R_6 = t_4^{r_{x_i}} g_1^{-r_\eta}. \end{cases}$$

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 $Sig = (t_1, t_2, T_3, t_4, c, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$.

Figure 3: The group signature generation algorithm SigGen

1. Randomly choose $s \in \mathbb{Z}_q^*$ and compute $C_1 = g_1^s, SK_1 = H_2(\hat{e}(u_2^s, H_0(ID_{\mathcal{L}_k})))$, where SK_1 will be the key of the symmetric key encryption scheme $\mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot)$.
2. Set $m_1 = Des||Add||TP||Sig$, and compute $C_2 = \mathcal{E}_{SK_1}(m_1)$, where Sig is the signature on $m_0 = Des||Add||TP$ which is generated using the SigGen algorithm in Figure 3.
3. Randomly choose $t \in \mathbb{Z}_q^*$ and compute $C_3 = g_1^t, SK_2 = H_2(\hat{e}(u_2^t, H_0(ID_{\mathcal{R}_j})))$.
4. Set $cipher_1 = C_1||C_2$, $m_2 = TP||ID_{\mathcal{L}_k}||cipher_1$, and compute $C_4 = \mathcal{E}_{SK_2}(m_2)$.
5. Send $cipher_2 = (C_3||C_4)$ to $\mathcal{R}_j$.

Step 2: When $\mathcal{R}_j$ receives $cipher_2 = (C_3, C_4)$, it does the following:

1. Compute $SK_2 = H_2(\hat{e}(C_3, S_{\mathcal{R}_j}))$.
2. Compute $m_2 = TP||ID_{\mathcal{L}_k}||cipher_1 = \mathcal{D}_{SK_2}(C_4)$.
3. Check TP to decide whether the request is fresh. If it is, send $cipher_1 = (C_1, C_2)$ to $\mathcal{L}_k$; otherwise abort.

Step 3: When $\mathcal{L}_k$ receives $cipher_1 = (C_1, C_2)$, it does the following:

1. Compute

$$\begin{cases} R_1 = g_1^{s_\alpha} u_0^{s_\beta} t_2^{-c} \\ R_2 = t_2^{s_{x_i}} g_1^{-s_\delta} u_0^{-s_\zeta} \\ R_3 = \hat{e}(t_1, g_2^{-s_{x_i}} u_3^{-c}) Y_0^{s_\alpha} Y_1^{s_\delta} Y_3^{c} \\ R_4 = Y_2^{s_\eta} T_3^{-c} \\ R_5 = g_1^{s_\gamma} t_4^{-c} \\ R_6 = t_4^{s_{x_i}} g_1^{-s_\eta}. \end{cases}$$

2. Check $c \stackrel{?}{=} H_1(\Psi, m_0, t_1, t_2, T_3, t_4, R_1, ..., R_6)$.
3. Output *valid* if the signature passes the above check.

Figure 4: The group signature verification algorithm SigVer

1. Compute $SK_1 = H_2(\hat{e}(C_1, S_{\mathcal{L}_k}))$.
2. Compute $m_1 = Des||Add||TP||Sig = \mathcal{D}_{SK_1}(C_2)$. If TP is fresh, go to next step; otherwise, abort.
3. Extract $Sig = (t_1, t_2, T_3, t_4, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$.
4. Check whether Sig is a valid group signature on $Des||Add||TP$ using the SigVer algorithm in Figure 4. If the signature is valid and the request is not from a revoked vehicle, provide the service to Add according to Des[2]; otherwise, abort.

[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 $T_3 \stackrel{?}{=} e(t_4, v_j)$ for all $v_j \in$ 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

[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 broadcast with the identifier, so that only the vehicle who generated the request needs to decrypt the encrypted outcome and any other vehicles cannot read the content of the outcome.

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 hierarchical approaches 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 $T_3 \stackrel{?}{=} e(t_4, v_i)$ for each tuple $(ID_{\mathcal{V}_i}, w_i, x_i, v_i)$ on RL. If this equation holds, KGC outputs $ID_{\mathcal{V}_i}$.

5. Evaluation

5.1. Security Analysis

In this section, we analyze the security of the LBS protocol. In the protocol, the underlying cryptographic primitives are the verifier-local revocation group signature scheme in [23], the Boneh and Franklin identity-based encryption (IBE) scheme in [8] and a secure symmetric-key encryption scheme (e.g., AES). The security of all these cryptographic tools is well established. On this solid security basis, we show that the security requirements listed in Section 2.2 are guaranteed.

It will be shown that the proposed protocol is secure against both external attackers and internal attackers (routing RSUs or LBS providers) defined in Section 2.2. This means that neither type of attacker can violate the privacy of a vehicle. For an external attacker, we require that he cannot learn the content of an LBS request or the service type that a vehicle is requesting or the service provided by an LBS provider. On the other hand, a curious routing RSU asked to submit an LBS request on behalf of the vehicle clearly needs to know the requested service type in order to forward the request to the corresponding LBS provider. However, we require that such an RSU cannot learn the content of the LBS request. Lastly, an LBS provider needs to know the service type/content of an LBS request in order to answer it correctly. However, we require the LBS provider cannot learn the vehicle's identity and cannot decide whether two different requests were generated by the same vehicle. Next, we show that our protocol meets the above security requirements in each step.

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

- *Level 1 message confidentiality and vehicle privacy* concerns external attackers. At this step, the secret key SK_2 used in the symmetric-key encryption scheme is protected by using the basic Boneh and Franklin [8] identity-based encryption (IBE) scheme which was proven to satisfy message confidentiality. An attacker cannot learn the value of SK_2 and, without SK_2, he cannot gain any meaningful information on the value of m_2; indeed, to gain such information, the attacker would need to break the provably secure Boneh-Franklin IBE scheme or the symmetric-key encryption scheme. Furthermore, in each session of the protocol, a random value t is chosen, which implies that, in each session, SK_2, C_3 and C_4 are different and independent from those in other sessions. Thus, the level 1 message confidentiality and vehicle privacy naturally follow.

- *Level 2 message confidentiality and vehicle privacy* concerns attacks from malicious or corrupted RSUs. In our protocol, the content of the LBS request is encrypted using the symmetric-key encryption scheme under the secret key SK_1. SK_1 is encrypted under the LBS provider's identity using the Boneh and Franklin IBE scheme [8]. Only the designated LBS provider who owns the private key corresponding to this identity can recover SK_1 and then use SK_1 to recover m_1. Thus, even if the private key of the RSU is leaked to the attacker, unless the attacker can break the Boneh and Franklin IBE scheme or the symmetric-key encryption scheme, he cannot gain any meaningful information on the content of the LBS request. Moreover, 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 this way, level 2 message confidentiality and vehicle privacy follow.

In Step 2, we have to cope with the attacks from both external attackers and malicious RSUs. Clearly, 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 type 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 $cipher_1$ without breaking the Boneh and Franklin IBE scheme or the symmetric-key encryption scheme. Similarly, an external attacker cannot learn the content of $cipher_1$ either, since he does not know the private key of the LBS provider. Therefore, the message confidentiality requirement is also met in this step. On the other hand, $cipher_1$ is also generated under the Boneh and

Franklin IBE scheme. The vehicle privacy requirement in this step follows accordingly.

Finally, we show that our protocol meets vehicle authentication, vehicle privacy, vehicle traceability and message confidentiality at Step 3 as 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 [23], then the LBS provider is convinced that the request comes from a registered vehicle. Hence, the required vehicle authentication is satisfied. Furthermore, the KGC can recover the identity of the vehicle, so vehicle traceability is also satisfied. As to vehicle privacy, since anyone can generate $Des||Add||TP$, it is easy to see that $Des||Add||TP$ cannot help the LBS provider to trace a vehicle. It remains to see whether Sig can be used by the LBS provider to trace a vehicle. However, the verifier-local revocation group signature in [23] 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 group member. Hence, Sig cannot help the LBS provider to trace a vehicle. As to message confidentiality, the service provided by an LBS provider is encrypted by using the symmetric-key encryption scheme; only the authenticated vehicles can decrypt the encrypted service. Thus, message confidentiality is achieved in this step.

We note that different types of attackers may collude. We show that no collusion can obtain more information about a request than either a routing RSU or the LBS provider alone, which reduces collusion attacks to internal attacks. If an external attacker colludes with a routing RSU, they can learn no more than the routing RSU, since the Boneh and Franklin IBE scheme is secure and without the private key of the LBS provider they cannot learn the content of an LBS request and hence cannot violate the privacy of a vehicle. Similarly, if an external attacker colludes with an LBS provider, they can learn no more than the LBS provider, since the underlying verifier-local revocation group signature scheme is secure. Further, if a routing RSU colludes with the corresponding LBS provider, or an external attacker colludes with both a routing RSU and the corresponding LBS provider, they can learn no more than the LBS provider. On the other hand, since the verifier-local revocation group signature is proven to be unforgeable and unlinkable, the colluders cannot forge a signature or distinguish whether two signatures come from the same sender (hence no profiling is possible).

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 assess the length of $C_3||TP||ID_{\mathcal{L}}||C_1||TP||Sig$. According to [9] and [23], the length of a point in $\mathbb{G}_1$ 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 [3], the minimal data rate in DSRC is 6 Mbps. Hence, we have that the maximal transmission delay caused by the security and privacy mechanism at Step 1 is about $\frac{434\times 8}{6\times 1024\times 1024}$ s ≈ 0.55 ms. This delay is very short and affordable for vehicles in VANETs.

5.3. Experimental Results

The efficiency of our system is mainly dominated by the LBS protocol stage and the revocation stage. In this section, we perform simulations to evaluate the efficiency of these two stages. The simulations were run on a Linux laptop using an Intel Core i5-2430M at a frequency of 2.4 GHz. In addition, we used the cryptographic library MIRACL [4] and implemented a MNT curve of embedded degree $k = 6$ and 160-bit q with a C program. The bilinear map was constructed with the Eta pairing.

In the LBS protocol stage, 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 4.34 ms. In the second step, for each LBS request, an RSU only needs to compute one pairing operation to decrypt the ciphertext $cipher_2$. The time is about 2.17 ms. Usually, the number of LBS requests received by an RSU in a short period of time will not be large and the RSU will be able to decrypt the ciphertexts in real time, because: i) the cover range of an RSU is not large; ii) the number of vehicles within the cover range is limited; iii) only some of the vehicles will submit an LBS request. In the last step, for each LBS request, the LBS provider needs to calculate two pairing operations to decrypt a ciphertext and verify the validity of a group signature. The computational cost of this step is shown in Figure 5. When only one LBS request is received, the time is about 8.49 ms. The time climbs to 802.17 ms when an LBS provider receives 100 requests in a short period of time. We

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

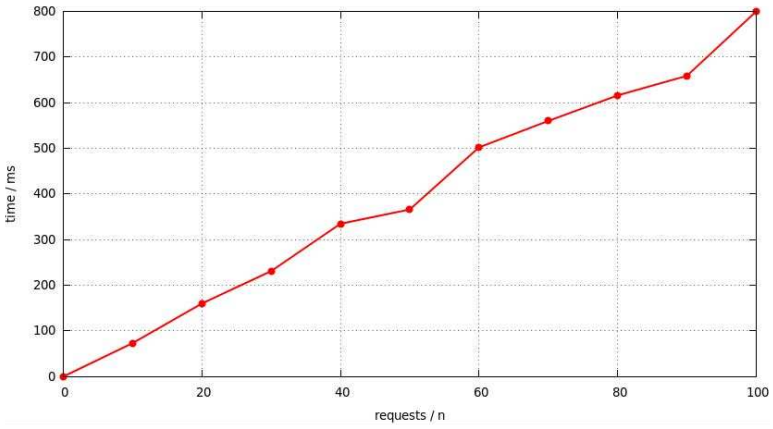

Figure 5: Cost of ciphertext decryption and group signature verification

note that such bursts only happen when the scale of a VANET is very large; yet the time remains affordable for most LBS applications. For some real-time LBS applications, the LBS provider may employ several sub-servers to decrypt and verify the messages. With this method, the time can be reduced to about $8.49n/l$ ms, where n is the number of requests received and l is the number of the sub-servers.

In the revocation stage, an LBS provider may receive many LBS requests from the vehicles in a very short period of time. Assume an LBS provider should verify n signatures at the same time and the revocation tokens in the revocation list RL are $(v_1, ..., v_\varphi)$. If the LBS provider checks whether an LBS request comes from a revoked vehicle token by token, then the LBS provider needs to compute $n\varphi$ pairing operations. Since a pairing operation is much more expensive than other operations in the system, the efficiency of the system declines dramatically as the number of revocation tokens in RL increases. Fortunately, revocation checking can be improved by using the batch verification technique in [7, 37]. With this batch technique, a vehicle can check whether a revocation token corresponds to one of the n vehicles. Let the n signatures be $Sig_1, ..., Sig_n$, where $Sig_j = (t_{j1}, t_{j2}, T_{j3}, t_{j4}, s_{j\alpha}, s_{j\beta}, s_{j\gamma}, s_{jx_i}, s_{j\delta}, s_{j\zeta}, s_{j\eta})$. The LBS provider randomly chooses ς_j of (a small number of) 80 bits and checks $\prod_{j=1}^{n} T_{j3}^{\varsigma_j} \stackrel{?}{=} e(\prod_{j=1}^{n} t_{j4}^{\varsigma_j}, v_l)$ for $1 \leqslant l \leqslant \varphi$. Using the batch verification technique, only φ (rather than $n\varphi$) pairing operations are required.

Figure 6 shows the relationship among the number of LBS requests received by an LBS provider, the number of tokens in RL and the cost to

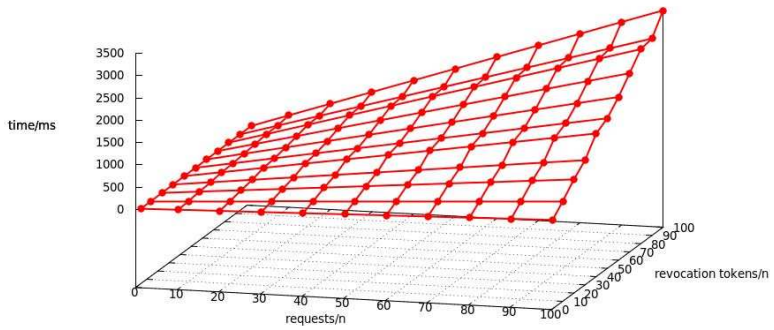

Figure 6: Cost of revocation checking

perform a revocation checking. When there is only one token in RL, the cost of revocation checking grows from 2.41 ms to 31.1 ms as the number of requests received by an LBS provider increases from 1 to 100. Therefore, the number of requests received by an LBS provider does not affect a lot the efficiency of the revocation checking. However, as the number of revocation tokens in the revocation list grows, the cost of revocation checking increases rapidly. When there are 100 tokens in RL, the cost grows from 238.5 ms to 3092 ms as the number of requests received by an LBS provider increases from 1 to 100. In order to avoid too large a cost for performing a revocation checking, one method is to take a small threshold τ. However, if τ is too small, the KGC has to update its MEK and corresponding public key, and re-issue member keys for all the vehicles with high frequency. The second method is similar to that in the above paragraph. The LBS provider may employ several revocation checking servers to speed up the revocation checking procedure. To operate, the tokens are separated into ℓ groups, where ℓ is the number of the servers, and the i-th server checks $\prod_{j=1}^{n} T_{j3}^{\varsigma_j} \stackrel{?}{=} e(\prod_{j=1}^{n} t_{j4}^{\varsigma_j}, v_l)$ for $(i-1)n/\ell + 1 \leqslant l \leqslant in/\ell$. With this arrangement, the time to check whether the requests come from legitimate vehicles can be reduced about ℓ times.

6. Extensions

6.1. Hierarchical KGCs and Multi-Issue Keys

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 generate private keys or member

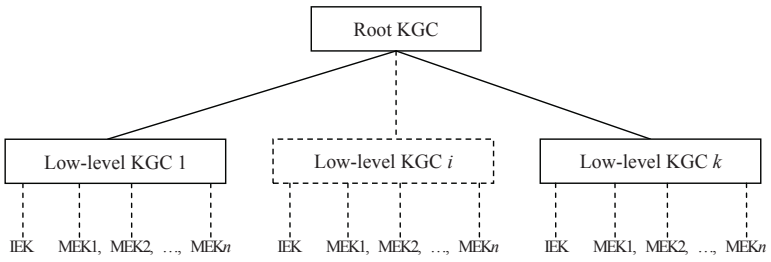

Figure 7: Hierarchical KGC and multi-issue key

keys for a large number of users, but also 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 keep the system efficient even if a large number of vehicles are revoked, we introduce an approach referred to as hierarchical KGCs and multi-issue keys (HKMK). The idea of HKMK is illustrated in Figure 7.

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, the latter can only learn that the vehicle belongs to a specific 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]

At this stage, the root KGC initializes the system-wide parameters. It does the following:

1. Choose three multiplicative cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ of the same order q, so that there exists a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $\mathbb{G}$ is generated by g_1 and $\mathbb{G}_2$ is generated by g_2.
2. Choose $u_0 \in \mathbb{G}_1, u_1 \in \mathbb{G}_2$ and compute $Y_1 = \hat{e}(u_0, g_2), Y_2 = \hat{e}(g_1, u_1), Y_3 = \hat{e}(g_1, g_2)$.

3. Choose a symmetric-key encryption scheme $\mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot)$.
4. Select cryptographic hash functions $H_0(\cdot) : \{0,1\}^* \longrightarrow \mathbb{G}_2$, $H_1(\cdot) : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$ and $H_2(\cdot) : \mathbb{G}_T \longrightarrow \{0,1\}^\lambda$.
5. Publish the system-wide parameters $\Psi = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, u_0, u_1, H_0, H_1, H_2, Y_1, Y_2, Y_3, \mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot))$. Ψ is assumed to be pre-loaded in each low-level KGC, vehicle, RSU and LBS provider.

One can notice that the root KGC no longer needs to maintain a member list and a revocation list. Instead, the low-level KGCs maintain their respective member and revocation lists.

[Low-Level KGC Setup]

After getting the system-wide parameters, a low-level KGC $\mathcal{K}_\mu$ generates its own parameters as follows:

1. Pick $\kappa_\mu \in Z_q^*$ as its identity enrolment key (IEK) and n member enrolment keys (MEKs) $\rho_{\mu 1}, ..., \rho_{\mu n} \in Z_q^*$.
2. Compute $u_{2\mu} = g_1^{\kappa_\mu}$ and $u_{3\mu i} = g_2^{\rho_{\mu i}}, 1 \leq i \leq n$ as its master public key.
3. Compute $Y_{0\mu i} = \hat{e}(u_0, u_{3\mu i}), 1 \leq i \leq n$.
4. Publish the parameters as $\Omega_\mu = (\mathcal{K}_\mu, u_{2\mu}, u_{3\mu 1}, ..., u_{3\mu n}, Y_{0\mu 1}, ..., Y_{0\mu n})$, where $\mathcal{K}_\mu$ is the identifying information of the low-level KGC.

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 $u_{3\mu j}$ and $Y_{0\mu j}$, and re-issues member keys for all the vehicles in the j-th sub-group.

6.2. Extended HKMK

In the above HKMK system, if an LBS provider records all the LBS requests received, then if a vehicle is revoked, an LBS provider can link the previous LBS requests of the revoked vehicle by the testing in the revocation stage. In other words, the privacy of the revoked vehicle is violated. To alleviate this problem, here we propose the extended HKMK technology. In the extended HKMK, the concept of time intervals is adopted. Each

vehicle has T revocation tokens corresponding to T time intervals, respectively. When a vehicle is revoked at interval i, signatures generated by the vehicle before interval i remain unlinkable. Therefore, the extended HKMK effectively protects the privacy of the vehicles in the system. The concrete system is proposed below.

[Root KGC Setup]

1. Choose three cyclic multiplicative groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ of the same order q, so that there exists a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $\mathbb{G}_1$ is generated by g_1 and $\mathbb{G}_2$ is generated by g_2.
2. Choose $u_0 \in \mathbb{G}_1; u_{11}, ..., u_{1T} \in \mathbb{G}_2$.
3. Compute $Y_1 = \hat{e}(u_0, g_2), Y_{21} = \hat{e}(g_1, u_{11}), ..., Y_{2T} = \hat{e}(g_1, u_{1T}), Y_3 = \hat{e}(g_1, g_2)$.
4. Choose a symmetric-key encryption scheme $\mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot)$ and the number of time intervals T.
5. Select cryptographic hash functions $H_0(\cdot) : \{0,1\}^* \longrightarrow \mathbb{G}_2$, $H_1(\cdot) : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$ and $H_2(\cdot) : \mathbb{G}_T \longrightarrow \{0,1\}^\lambda$.
6. Publish the system-wide parameters $\Psi = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, u_0, u_{11}, ..., u_{1T}, H_0, H_1, H_2, Y_1, Y_{21}, ..., Y_{2T}, Y_3, \mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot), T)$. 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 $\mathcal{K}_\mu$ generates its own parameters as follows:

1. Pick $\kappa_\mu \in Z_q^*$ as its identity enrolment key (IEK) and n member enrolment keys (MEKs) $\rho_{\mu 1}, ..., \rho_{\mu n} \in Z_q^*$.
2. Compute $u_{2\mu} = g_1^{\kappa_\mu}$ and $u_{3\mu i} = g_2^{\rho_{\mu i}}$ for $1 \leq i \leq n$.
3. Compute $Y_{0\mu i} = \hat{e}(u_0, u_{3\mu i}), 1 \leq i \leq n$.
4. Publish the parameters as $\Omega_\mu = (\mathcal{K}_\mu, u_{2\mu}, u_{3\mu 1}, ..., u_{3\mu n}, Y_{0\mu 1}, ..., Y_{0\mu n})$.

The low-level KGC $\mathcal{K}_\mu$ also maintains n member lists $\mathsf{ML}_1, ..., \mathsf{ML}_n$ and nT revocation lists $\mathsf{RL}_{1,1}, ..., \mathsf{RL}_{1,T}; ...; \mathsf{RL}_{n1}, ..., \mathsf{RL}_{nT}$. To deal with the revocation problem more efficiently, similarly to our basic system, $\mathcal{K}_\mu$ chooses a threshold τ. If a vehicle $\mathcal{V}_\nu$ is compromised at interval l (we assume the member key of $\mathcal{V}_\nu$ is issued by using the j-th MEK of $\mathcal{K}_\mu$, $1 \leq j \leq n$) and the number of tokens in RL_{jT} is not greater than τ, $\mathcal{K}_\mu$ first finds the revocation tokens $v_l, ..., v_T$ of $\mathcal{V}_\nu$ in ML_j, then adds the revocation token v_s to RL_{js}, $l \leq s \leq T$. Otherwise, $\mathcal{K}_\mu$ updates its j-th MEK, the corresponding public

key $u_{3\mu j}$ and $Y_{0\mu j}$, and re-issues member keys for all the vehicles in the j-th sub-group.

[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 $ID_{\mathcal{R}_i}$ and $\mathcal{R}_i$ will be enrolled by the low-level KGC whose IEK is $\kappa_\mu \in Z_q^*$. The low-level KGC computes $S_{\mathcal{R}_i} = H_0(ID_{\mathcal{R}_i})^{\kappa_\mu}$.

ServiceJoin: This algorithm is used to generate the secret key for an LBS provider. Suppose that the identity of an LBS provider is $ID_{\mathcal{L}_i}$ and $\mathcal{L}_i$ will be enrolled by the low-level KGC whose IEK is $\kappa_\mu \in Z_q^*$. The low-level KGC computes $S_{\mathcal{L}_i} = H_0(ID_{\mathcal{L}_i})^{\kappa_\mu}$.

VehicleJoin: This algorithm is used to generate the member keys for a vehicle. The low-level KGC maintains n member lists of tuples $(ID_{\mathcal{V}}, w, x, v_1, ..., v_T)$. When a vehicle wants to join the system, the low-level KGC whose MEKs are $\rho_{\mu 1}, ..., \rho_{\mu n} \in Z_q^*$ accepts a vehicle's identity $ID_{\mathcal{V}_i}$ and generates the member key as follows:

1. Select $x_i \in \mathbb{Z}_q^*$.
2. Randomly select a MEK $\rho_{\mu j}$, compute $w_i = g_1^{1/(\rho_{\mu j}+x_i)}$ and set (w_i, x_i) as the member key of $\mathcal{V}_i$, where $1 \leq j \leq n$.
3. Add $(ID_{\mathcal{V}_i}, w_i, x_i, v_{i1}, ..., v_{iT})$ to ML_j, where $v_{il} = u_{1l}^{x_i}, 1 \leq l \leq T$, is the revocation token of $\mathcal{V}_i$.

[LBS Protocol]

This stage is similar to that in Section 4. The LBS protocol consists of the following three steps.

Step 1: The first step is the vehicle-to-RSU communication. Suppose that $\mathcal{V}_i$ with member key (w_i, x_i) wants to access the LBS provider $\mathcal{L}_j$ who is enrolled by the low-level KGC $\mathcal{K}_\mu$ and its nearby RSU is $\mathcal{R}_k$ who is enrolled by the low-level KGC $\mathcal{K}_\nu$. $\mathcal{V}_i$ is enrolled by the low-level KGC $\mathcal{K}_\vartheta$ using its l-th MEK. Assume the current interval is θ. $\mathcal{V}_i$ does the following:

1. Choose $s \in \mathbb{Z}_q^*$ and compute $C_1 = g_1^s$, $SK_1 = H_2(\hat{e}(u_{2\mu}^s, H_0(ID_{\mathcal{L}_j})))$, where SK_1 will be the key of the symmetric-key encryption scheme $\mathcal{E}_K(\cdot)/\mathcal{D}_K(\cdot)$.

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 $t_1 = w_i u_0^{\alpha}, t_2 = g_1^{\alpha} u_0^{\beta}, T_{3\theta} = Y_{2\theta}^{\eta}$ and $t_4 = g_1^{\gamma}$.
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

$$\begin{cases} R_1 = g_1^{r_\alpha} u_0^{r_\beta} \\ R_2 = t_2^{r_{x_i}} g_1^{-r_\delta} u_0^{-r_\zeta} \\ R_3 = \hat{e}(t_1, g_2)^{-r_{x_i}} Y_{0\vartheta l}^{r_\alpha} Y_1^{r_\delta} \\ R_4 = Y_{2\theta}^{r_\eta} \\ R_5 = g_1^{r_\gamma} \\ R_6 = t_4^{r_{x_i}} g_1^{-r_\eta}. \end{cases}$$

5. Compute $c = H_1(\Psi, \mathcal{K}_\vartheta, l, \theta, m_0, t_1, t_2, T_{3\theta}, t_4, R_1, ..., R_6)$.
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 $Sig = (\mathcal{K}_\vartheta, l, \theta, t_1, t_2, T_{3\theta}, t_4, c, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$.

Figure 8: The group signature generation algorithm SigGen2

2. Set $m_1 = Des||Add||TP||Sig$, and compute $C_2 = \mathcal{E}_{SK_1}(m_1)$, where Sig is the signature on $m_0 = Des||Add||TP$ which is generated using the SigGen2 algorithm in Figure 8.
3. Choose $t \in \mathbb{Z}_q^*$ and compute $C_3 = g_1^t$, $SK_2 = H_2(\hat{e}(u_{2\nu}^t, H_0(ID_{\mathcal{R}_k})))$.
4. Set $cipher_1 = C_1||C_2$, $m_2 = TP||ID_{\mathcal{L}_j}||cipher_1$, and compute $C_4 = \mathcal{E}_{SK_2}(m_2)$.
5. Send $cipher_2 = (C_3||C_4)$ to $\mathcal{R}_k$.

Step 2: When $\mathcal{R}_k$ receives $cipher_2 = (C_3, C_4)$, it does the following:

1. Compute $SK_2 = H_2(\hat{e}(C_3, S_{\mathcal{R}_k}))$.
2. Compute $m_2 = TP||ID_{\mathcal{L}_j}||cipher_1 = \mathcal{D}_{SK_2}(C_4)$.
3. Check TP to decide whether the request is fresh. If it is, send $cipher_1 = (C_1, C_2)$ to $\mathcal{L}_j$; otherwise abort.

Step 3: When $\mathcal{L}_j$ receives $cipher_1 = (C_1, C_2)$, it does the following:

1. Compute $SK_1 = H_2(\hat{e}(C_1, S_{\mathcal{L}_j}))$.
2. Compute $m_1 = Des||Add||TP||Sig = \mathcal{D}_{SK_1}(C_2)$. If TP is fresh, go to next step; otherwise, abort.

1. Compute

$$\begin{cases} R_1 = g_1^{s_\alpha} u_0^{s_\beta} t_2^{-c} \\ R_2 = t_2^{s_{x_i}} g_1^{-s_\delta} u_0^{-s_\zeta} \\ R_3 = \hat{e}(t_1, g_2^{-s_{x_i}} u_{3\vartheta l}^{-c}) Y_{0\vartheta l}^{s_\alpha} Y_1^{s_\delta} Y_3^{c} \\ R_4 = Y_{2\theta}^{s_\eta} T_{3\theta}^{-c} \\ R_5 = g_1^{s_\gamma} t_4^{-c} \\ R_6 = t_4^{s_{x_i}} g_1^{-s_\eta}. \end{cases}$$

2. Check $c \stackrel{?}{=} H_1(\Psi, \mathcal{K}_\vartheta, l, \theta, m_0, t_1, t_2, T_3, t_4, R_1, ..., R_6)$.
3. Output *valid* if the signature passes the above check.

Figure 9: The group signature verification algorithm SigVer2

3. Extract $Sig = (\mathcal{K}_\vartheta, l, \theta, t_1, t_2, T_{3\theta}, t_4, c, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$.
4. Check whether Sig is a valid group signature on $Des||Add||TP$ using the SigVer2 algorithm in Figure 9. If the signature is valid, provide the service to Add according to Des.

[Revocation]

To tackle the revocation problem, when a vehicle $\mathcal{V}_i$ who is enrolled by the low-level KGC $\mathcal{K}_\vartheta$ using its l-th MEK is compromised at time interval θ, $\mathcal{K}_\vartheta$ finds the corresponding $(ID_{\mathcal{V}_i}, w_i, x_i, v_{i1}, ..., v_{iT})$ on ML_l, and then adds the revocation token $v_{i\pi}$ to $\mathsf{RL}_{l\pi}$, for $l \leq \pi \leq T$. To detect whether a group signature $Sig = (\mathcal{K}_\vartheta, l, \theta, t_1, t_2, T_{3\theta}, t_4, c, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$ is generated by a revoked vehicle, the LBS provider checks $T_{3\theta} \stackrel{?}{=} e(t_4, v_{j\theta})$ for all $v_{j\theta} \in \mathsf{RL}_{l\theta}$. If none of the equations holds, it means that the vehicle is not revoked. We note that the techniques used in Section 5.3 are also applicable here to speed up the revocation checking. When there are too many revoked vehicles in RL_{lT}, we may allow $\mathcal{K}_\vartheta$ to choose a threshold τ; and when the number of revoked vehicles in RL_{lT} is greater than τ, $\mathcal{K}_\vartheta$ updates its l-MEK and corresponding public key, and re-issues member keys for all the vehicles in its group.

[Trace]

Let $Sig = (\mathcal{K}_\vartheta, l, \theta, t_1, t_2, T_{3\theta}, t_4, c, s_\alpha, s_\beta, s_\gamma, s_{x_i}, s_\delta, s_\zeta, s_\eta)$ be a valid group signature. To trace a vehicle, the low-level KGC $\mathcal{K}_\vartheta$ checks $T_{3\theta} \stackrel{?}{=} e(t_4, v_{i\theta})$

for the tuple $(ID_{\mathcal{V}_i}, w_i, x_i, v_{i1}, ..., v_{i1})$ on ML_l. If this equation holds, $\mathcal{K}_\vartheta$ outputs $ID_{\mathcal{V}_i}$.

7. Conclusion

In this paper, we have proposed a new location based service protocol that efficiently addresses the security and conditional privacy challenges inherent to offering LBSs in VANETs. In our system, both RSUs and LBS providers are identity-based, and a vehicle only needs a member key. With its member key, a vehicle can generate verifier-local revocation group signatures. Those signatures can be validated by the LBS providers without violating the privacy of the vehicles. Furthermore, if an LBS request is found to be false, the key generation center can determine the identity of the vehicle. Our analysis shows that our scheme is a practical one.

Acknowledgments and disclaimer

This work was supported in part by the the NSF of China under grants 61202465, 61321064, 91018008, 60970114, 60970115, 60970116, 61173154, 61003214, 61173192, 61103222; the European Commission under projects FP7 "DwB", FP7 "Inter-Trust" and H2020 "CLARUS"; the Spanish Government under project TIN2011-27076-C03-01; the Government of Catalonia under grant 2014 SGR 537; the Templeton World Charity Foundation under grant "CO-UTILITY"; the Shanghai NSF under Grant No. 12ZR1443500; the Shanghai Chen Guang Program (12CG24); the Fundamental Research Funds for the Central Universities of China. J. Domingo-Ferrer was supported in part as an ICREA-Acadèmia researcher by the Government of Catalonia. The views presented here do not necessarily reflect those of UNESCO, the Templeton World Charity Foundation or any of the funders.

References

[1] Car 2 Car Communication Consortium. `http://www.car-to-car.org/`

[2] European Parliament. Legislative resolution on the proposal for a directive of the European Parliament and of the Council on the retention of data processed in connection with the provision of public electronic communication services and amending Directive 2002/58/EC (COM(2005)0438 C6-0293/2005 2005/0182(COD)), 2005.

[3] Dedicated Short Range Communications (DRSC) home. http://www.leearmstrong.com/Dsrc/DSRCHomeset.htm

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

[5] Mastercard and Visa. SET protocol specifications, 1997. http://www.setco.org/set_specifications.html

[6] M. Bellare, J. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. V. Herreweghen and M. Waidner. Design, implementation, and deployment of the iKP Secure Electronic Payment System. IEEE Journal of Selected Areas in Communications, vol. 18, no. 4, pp. 611-627, 2000.

[7] M. Bellare, J. Garay and T. Rabin. Fast batch verification for modular exponentiation and digital signatures. In Advances in Cryptology-EUROCRYPT 1998, Lecture Notes in Computer Science, vol. 1403, pp. 236-250, 1998.

[8] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In Advances in Cryptology-CRYPTO 2001, Lecture Notes in Computer Science, vol. 2139, pp. 213-229, 2001.

[9] D. Boneh, B. Lynn and H. Shacham. Short signatures from the Weil pairing. In Asiacrypt 2001, Lecture Notes in Computer Science, vol. 2248, pp. 514-532, 2001.

[10] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In ACM Conference on Computer and Communications Security, pp. 168-177, 2004.

[11] D. Chaum. Blind signatures for untraceable payments. In Advances in Cryptology-Crypto 1982, Plenum Press, pp. 199-203, 1983.

[12] D. Chaum and E. van Heijst. Group signatures. In Advances in Cryptology-Eurocrypt 1991, Lecture Notes in Computer Science, vol. 576, pp. 257-265, 1991.

[13] R. Engoulou, M. Bellaïche and S. Pierre, A. Quintero. VANET security surveys. Computer Communications, vol. 44, pp. 1-13, 2014.

[14] G. Frey and H.-G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. Mathematics of Computation, vol. 62, no. 206, pp. 865-874, 1994.

[15] J. Guo, J. P. Baugh and S. Wang. A group signature based secure and privacy-preserving vehicular communication framework. In 2007 Mobile Networking for Vehicular Environments, pp. 103-108, 2007.

[16] F. Hess, N.P. Smart and F. Vercauteren. The Eta pairing revisited. IEEE Transactions on Information Theory, vol. 52, no. 10, pp. 4595-4602, 2006.

[17] J. Huang, L. Yeh and H. Chien. ABAKA: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks, IEEE Transactions on Vehicular Technology, vol. 60, no. 1, pp. 248-262, 2011.

[18] J. Hubaux, S. Ĉapkun and J. Luo. The security and privacy of smart vehicles. IEEE Security and Privacy, vol. 2, no. 3, pp. 49-55, 2004.

[19] P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Advances in Cryptology-CRYPTO 1996, Lecture Notes in Computer Science, vol. 1109, pp. 104-113, 1996.

[20] X. Lin, X. Sun, P. Ho and X. Shen. GSIS: A secure and privacy preserving protocol for vehicular communications. IEEE Transactions on Vehicular Technology, vol. 56, no. 6, pp. 3442-3456, 2007.

[21] B. Liu and L. Zhang. An improved identity-based batch verification scheme for VANETs. In 5th International Conference on Intelligent Networking and Collaborative Systems (INCos 2013), pp. 809-814, 2013.

[22] A. Menezes, T. Okamoto and S.A. Vanstone. Reducing elliptic curves logarithms to logarithms in a finite field. IEEE Transactions on Information Theory, vol. 39, no. 5, pp. 1639-1646, 1993.

[23] T. Nakanishi and N. Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In Advances in Cryptology - ASIACRYPT 2005, Lecture Notes in Computer Science, vol. 3788, pp. 533-548, 2005.

[24] P. Papadimitratos, V. Gligor and J. Hubaux. Securing vehicular communications - Assumptions, requirements, and principles. In Workshop on Embedded Security in Cars (ESCAR) 2006, 2006.

[25] R. DiPietro, S. Guarino, N. Verde and J. Domingo-Ferrer. Security in wireless ad-hoc networks - A survey. Computer Communications, vol. 51, pp. 1-20, 2014.

[26] M. Raya and J.-P. Hubaux. Security aspects of inter-vehicle communications. In Proc. of Swiss Transport Research Conference, March 2005.

[27] M. Raya and J. Hubaux. The security of vehicular ad hoc networks. In 3rd ACM Workshop on Security of Ad hoc and Sensor Networks-SASN 05, pp. 11-21, 2005.

[28] M. Raya and J. Hubaux. Securing vehicular ad hoc networks. Journal of Computer Security, vol. 15, no. 1, pp. 39-68, 2007.

[29] K. Sampigethaya, M. Li, L. Huang and R. Poovendran. AMOEBA: Robust location privacy scheme for VANET. IEEE Journal on Selected Areas in Communications, vol. 25, no. 8, pp. 1569-1589, 2007

[30] A. Shamir. Identity based cryptosystems and signature schemes. In Advances in Cryptology-CRYPTO 1984, Lecture Notes in Computer Science, vol. 196, pp. 47-53, 1984.

[31] F. Standaert, T. Malkin and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In Advances in Cryptology-Eurocrypt 2009, Lecture Notes in Computer Science, vol. 5479, pp. 443-461, 2009.

[32] H. Wang and Y. Zhang. On the security of an anonymous batch authenticated and key agreement scheme for value-added services in VANETs. Procedia Engineering, vol. 29, pp. 1735-1739, 2012.

[33] H. Yong, T. Jin, C. Yu and Z. Chi. Secure data downloading with privacy preservation in vehicular ad hoc networks. In IEEE International Conference on Communications 2010, pp. 1-5,2010.

[34] M. Zarki, S. Mehrotra, G. Tsudik and N. Venkatasubramanian. Security issues in a future vehicular network. In Proc. of the European Wireless Workshop, February 2002.

[35] C. Zhang, R. Lu, X. Lin, P. Ho and X. Shen. An efficient identity-based batch verification scheme for vehicular sensor networks. In IEEE INFOCOM 2008, pp. 246-250, 2008.

[36] Q. Wu, J. Domingo-Ferrer and U. González-Nicolás. Balanced trustworthiness, safety and privacy in vehicle-to-vehicle communications. IEEE Transactions on Vehicular Technology, vol. 59 no. 2, pp. 559-573, 2010.

[37] L. Zhang, Q. Wu, A. Solanas and J. Domingo-Ferrer. A scalable robust authentication protocol for secure vehicular communications. IEEE Transactions on Vehicular Technology, vol. 59, no. 4, 1606-1617, 2010.

[38] L. Zhang, Q. Wu, B. Qin and J. Domingo-Ferrer. Provably secure one-round identity-based authenticated asymmetric group key agreement protocol. Information Sciences, vol. 181, no. 19, 4318-4329, 2011.

[39] L. Zhang, Q. Wu, B. Qin and J. Domingo-Ferrer. Practical privacy for value-added applications in vehicular ad hoc networks. In IDCS 2012 and ICDKE 2012, Lecture Notes in Computer Science, vol. 7646, pp. 43-56, 2012.